Lei Wang
Michael Segal
Jenhui Chen
Tie Qiu (Eds.)

# Wireless Algorithms, Systems, and Applications

LNCS 13471

17th International Conference, WASA 2022
Dalian, China, November 24–26, 2022
Proceedings, Part I


Springer

# Lecture Notes in Computer Science 13471

More information about this series at https://link.springer.com/bookseries/558

Lei Wang · Michael Segal · Jenhui Chen ·
Tie Qiu (Eds.)

# Wireless Algorithms, Systems, and Applications

17th International Conference, WASA 2022
Dalian, China, November 24–26, 2022
Proceedings, Part I

Springer

*Editors*
Lei Wang
Dalian University of Technology
Dalian, China

Jenhui Chen
Chang Gung University
Taiwan, China

Michael Segal
Ben-Gurion University of the Negev
Beer-Sheva, Israel

Tie Qiu
Tianjin University
Tianjin, China

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-031-19207-4 ISBN 978-3-031-19208-1 (eBook)
https://doi.org/10.1007/978-3-031-19208-1

# Preface

The 17th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2022) was held in Dalian during November 24–26, 2022. The conference focused on new ideas and recent advances in computer systems, wireless networks, distributed applications, and advanced algorithms that are pushing forward the new technologies for better information sharing, computer communication, and universal connected devices in various environments, especially in wireless networks. WASA has become a broad forum for computer theoreticians, system and application developers, and other professionals in networking-related areas to present their ideas, solutions, and knowledge of emerging technologies and challenges in computer systems, wireless networks, and advanced applications.

The technical program of WASA 2022 consisted of 94 regular papers and 68 short papers, selected by the Program Committee from 265 full submissions in response to the call for papers. All submissions were reviewed by at least 115 Program Committee members in a 115 double blind process. The submissions cover numerous cutting edge topics: cognitive radio networks; software-defined radio and reconfigurable radio networks; cyber-physical systems (CPSs) including intelligent transportation systems and smart healthcare systems; theoretical frameworks and analysis of fundamental cross-layer protocol and network design and performance issues; distributed and localized algorithm design and analysis; information and coding theory for wireless networks; localization; mobility models and mobile social networking; mobile cloud; topology control and coverage; security and privacy; underwater and underground networks; vehicular networks; radar and sonar networks; PHY/MAC/routing protocols; information processing and data management; programmable service interfaces; energy-efficient algorithms; systems and protocol design; operating system and middleware support; algorithms, systems, and applications of the Internet of Things (IoT); and algorithms, systems, and applications of edge computing, etc. In the first place, we would like to express our grateful appreciation for all Program Committee members for their hard work in reviewing all submissions. Furthermore, we would like to give our special thanks to the WASA Steering Committee for their consistent leadership and guidance; also, we would like to extend our gratitude to the the local chairs (Jingang Yu, Zumin Wang, and Jie Wang), the publication chairs (Chi Lin, Lei Shu, Guangjie Han, and Pengfei Wang), the publicity chairs (Zichuan Xu, Haipeng Dai, Zhibo Wang, and Chenren Xu), organizing chairs (Dongsheng Zhou and Zhenquan Qin), and the Web chair (Bingxian Lu) for their remarkable contributions to WASA 2022, ensuring that it was a successful conference. In particular, we wish to express our deepest respect and

thankfulness to all the authors for submitting and presenting their outstanding ideas and solutions at the conference.

November 2022

Lei Wang
Michael Segal
Jenhui Chen
Tie Qiu

# Organization

## Steering Committee Members

| | |
|---|---|
| Xiuzhen Susan Cheng | George Washington University, USA |
| Zhipeng Cai | Georgia State University, USA |
| Jiannong Cao | Hong Kong Polytechnic University, Hong Kong, China |
| Ness Shroff | The Ohio State University, USA |
| Wei Zhao | University of Macau, Macau, China |
| Peng-Jun Wan | Illinois Institute of Technology, USA |
| Ty Znati | University of Pittsburgh, USA |
| Xinbing Wang | Shanghai Jiao Tong University, China |

## General Co-chairs

| | |
|---|---|
| Zhongxuan Luo | Dalian University of Technology, China |
| Peng-Jun Wan | Illinois Institute of Technology, USA |
| Xingwei Wang | Northeastern University, China |

## Program Co-chairs

| | |
|---|---|
| Lei Wang | Dalian University of Technology, China |
| Michael Segal | Ben-Gurion University, Israel |
| Jen-Hui Chen | Chang Gung University, Taiwan, China |
| Tie Qiu | Tianjin University, China |

## Publicity Co-chairs

| | |
|---|---|
| Zichuan Xu | Dalian University of Technology, China |
| Haipeng Dai | Nanjing University, China |
| Zhibo Wang | Zhejiang University, China |
| Chenren Xu | Peking University, China |

## Publication Co-chairs

| | |
|---|---|
| Chi Lin | Dalian University of Technology, China |
| Lei Shu | Nanjing Agricultural University, China |
| Guangjie Han | Hohai University, China |
| Pengfei Wang | Dalian University of Technology, China |

## Local Co-chairs

| | |
|---|---|
| Jingang Yu | University of Chinese Academy of Sciences, China |
| Zumin Wang | Dalian University, China |
| Jie Wang | Dalian Maritime University, China |

## Web Chair

| | |
|---|---|
| Bingxian Lu | Dalian University of Technology, China |

## Organizing Co-chairs

| | |
|---|---|
| Dongsheng Zhou | Dalian University, China |
| Zhenquan Qin | Dalian University of Technology, China |

## Program Committee

| | |
|---|---|
| Ran Bi | Dalian University of Technology, China |
| Edoardo Biagioni | University of Hawaii at Manoa, USA |
| Salim Bitam | University of Biskra, Algeria |
| Azzedine Boukerche | University of Ottawa, Canada |
| Zhipeng Cai | Georgia State University, USA |
| Srinivas Chakravarthi Thandu | Amazon, USA |
| Sriram Chellappan | University of South Florida, USA |
| Quan Chen | Guangdong University of Technology, China |
| Xianfu Chen | VTT Technical Research Centre of Finland, Finland |
| Xu Chen | Sun Yat-sen University, China |
| Wei Wang | Sun Yat-sen University, China |
| Songqing Chen | George Mason University, USA |
| Soufiene Djahel | Manchester Metropolitan University, UK |
| Yingfei Dong | University of Hawaii, USA |
| Zhuojun Duan | James Madison University, USA |
| Luca Foschini | University of Bologna, Italy |
| Jing Gao | Dalian University of Technology, China |
| Xiaofeng Gao | Shanghai Jiao Tong University, China |
| Jidong Ge | Nanjing University, China |
| Chunpeng Ge | Nanjing University of Aeronautics and Astronautics, China |
| Daniel Graham | University of Virginia, USA |
| Ding Wang | Nankai University, China |
| Ning Gu | Fudan University, China |

| | |
|---|---|
| Deke Guo | National University of Defense Technology, China |
| Bin Guo | Northwestern Polytechnical University, China |
| Meng Han | Kennesaw State University, USA |
| Suining He | University of Connecticut, USA |
| Zaobo He | Miami University, USA |
| Pengfei Hu | Shandong University, China |
| Peng Sun | The Chinese University of Hong Kong, China |
| Yan Huang | Kennesaw State University, USA |
| Yan Huo | Beijing Jiaotong University, China |
| Holger Karl | University of Paderborn, Germany |
| Donghyun Kim | Kennesaw State University, USA |
| Hwangnam Kim | Korea University, South Korea |
| Bharath Kumar Samanthula | Montclair State University, USA |
| Abderrahmane Lakas | United Arab Emirates University, UAE |
| Sanghwan Lee | Kookmin University, South Korea |
| Feng Li | Shandong University, China |
| Feng Li | Indiana University-Purdue University Indianapolis, USA |
| Ruinian Li | Bowling Green State University, USA |
| Wei Li | Georgia State University, USA |
| Zhenhua Li | Tsinghua University, China |
| Zhetao Li | Xiangtan University, China |
| Peng Li | University of Aizu, Japan |
| Qi Li | Tsinghua University, China |
| Yaguang Lin | Shaanxi Normal University, China |
| Zhen Ling | Southeast University, China |
| Weimo Liu | George Washington University, USA |
| Jia Liu | Nanjing University, China |
| Fangming Liu | Huazhong University of Science and Technology, China |
| Liang Liu | Beijing University of Posts and Telecommunications, China |
| Hongbin Luo | Beihang University, China |
| Jun Luo | Nanyang Technological University, Singapore |
| Liran Ma | Texas Christian University, USA |
| Jian Mao | Beihang University, China |
| Bo Mei | Texas Christian University, USA |
| Hung Nguyen | Carnegie Mellon University, USA |
| Pasquale Pace | University of Calabria, Italy |
| Claudio Palazzi | University of Padua, Italy |
| Chuan Lin | Northeastern University, China |

| | |
|---|---|
| Junjie Pang | Qingdao University, China |
| Javier Parra-Arnau | University of Ottawa, Canada |
| Tie Qiu | Tianjin University, China |
| Ruben Rios | University of Malaga, Spain |
| Kazuya Sakai | Tokyo Metropolitan University, Japan |
| Omar Sami Oubbati | University of Laghouat, Algeria |
| Kewei Sha | University of Houston - Clear Lake, USA |
| Hao Sheng | Beihang University, China |
| Bo Sheng | University of Massachusetts Boston, USA |
| Tuo Shi | Harbin Institute of Technology, China |
| Tong Liu | Shanghai University, China |
| Sukhpal Singh Gill | Queen Mary University of London, UK |
| Junggab Son | Kennesaw State University, USA |
| Riccardo Spolaor | Shandong University, China |
| Chunhua Su | University of Aizu, Japan |
| Violet Syrotiuk | Arizona State University, USA |
| Guoming Tang | National University of Defense Technology, China |
| Bin Tang | Hohai University, China |
| Xiaohua Tian | Shanghai Jiao Tong University, China |
| Luis Urquiza | Universitat Politècnica de Catalunya, Spain |
| Tian Wang | Huaqiao University, China |
| Yawei Wang | George Washington University, USA |
| Yingjie Wang | Yantai University, China |
| Zhibo Wang | Zhejiang University, China |
| Leye Wang | Peking University, China |
| Wei Wei | Xi'an University of Technology, China |
| Alexander Wijesinha | Towson University, USA |
| Mike Wittie | Montana State University, USA |
| Kaishun Wu | Shenzhen University, China |
| Xiaobing Wu | University of Canterbury, New Zealand |
| Wei Xi | Xi'an Jiaotong University, China |
| Yang Xiao | University of Alabama, USA |
| Kun Xie | Hunan University, China |
| Xuan Liu | Hunan University, China |
| Kaiqi Xiong | University of South Florida, USA |
| Kuai Xu | Arizona State University, USA |
| Wen Xu | Texas Woman's University, USA |
| Lei Yang | The Hong Kong Polytechnic University, China |
| Panlong Yang | University of Science and Technology of China, China |

| | |
|---|---|
| Changyan Yi | Nanjing University of Aeronautics and Astronautics, China |
| Wei Yu | Towson University, USA |
| Dongxiao Yu | Shandong University, China |
| Sherali Zeadally | University of Kentucky, USA |
| Deze Zeng | China University of Geosciences, China |
| Bowu Zhang | Marist College, USA |
| Yong Zhang | Shenzhen Institutes of Advanced Technology, China |
| Yang Zhang | Wuhan University of Technology, China |
| Cheng Zhang | George Washington University, USA |
| Xu Zheng | University of Science and Technology of China, China |
| Yanwei Zheng | Shandong University, China |
| Lu Zhou | Nanjing University of Aeronautics and Astronautics, China |
| Jindan Zhu | Amazon Web Services, USA |
| Tongxin Zhu | Southeast University, China |
| Yifei Zou | Shandong University, China |

# Contents – Part I

**Security and Privacy**

**Energy-Efficient Algorithms, Systems and Protocol Design**

# Contents – Part II

# Contents – Part III

**Vehicular Networks**



**PHY/MAC/Routing Protocols**

**Algorithms, Systems, and Applications of Edge Computing**

# Cyber-Physical Systems Including Intelligent Transportation Systems and Smart Healthcare Systems

# An Efficient Privacy-Preserving Scheme for Traffic Monitoring Services in Vehicular Networks

Chen Gu, Xuande Cui, and Donghui Hu(✉)

School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, Anhui, China
{guchen,hudh}@hfut.edu.cn, xuandecui@mail.hfut.edu.cn

**Abstract.** Traffic monitoring services show great potential for improving the traffic efficiency in vehicular networks. Drivers can obtain the latest information on their upcoming routes by sending location-based queries to the traffic monitoring server. However, it is inefficient to query all events on the route due to the timeliness of traffic events. Besides, sending location-based queries will expose drivers' privacy. Existing research does not consider both issues under the traffic monitoring scenario. In this paper, we propose an efficient privacy-preserving scheme that ensures privacy is protected in two dimensions. Both location privacy and identity privacy are preserved, such that attackers cannot observe the real location as well as the identity via continuous queries. Specifically, the scheme first segments the whole route into multiple ones. Then drivers send endpoints in each segmented route to satisfy the timeliness requirement. We propose a location obfuscation mechanism based on *geo-indistinguishability* and utilize it on every segmented route. We address an issue in geo-indistinguishability where the obfuscated location is unreasonable. Additionally, continuous queries may expose the driver's identity. We thus define a type of attack called *identity linking attack* and propose two possible solutions. We finally conduct experiments on the real dataset. Experimental results demonstrate the efficiency of our proposed scheme.

**Keywords:** Location privacy · Identity linking attack · Vehicular networks · Traffic monitoring services

## 1 Introduction

The rapid development of vehicular networks in recent years shows great potential for improving traffic efficiency. In the traffic monitoring applications, vehicles

This work was supported in part by Anhui Science and Technology Key Special Program under Grant No. 201903a05020016, in part by the National Natural Science Foundation of China (NSFC) under Grant No. U1836102, in part by Anhui Provincial Natural Science Foundation under Grant No. 2008085MF196, and in part by the Fundamental Research Funds for the Central Universities under Grant No. JZ2022HGQA0166.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 3–14, 2022.
https://doi.org/10.1007/978-3-031-19208-1_1

equipped with on-board units (OBU) and a variety of sensors (*e.g.,* multi-beam lidars and high-resolution cameras) can sense the surrounding environments and report events such as car accidents and traffic congestion to the cloud server [15]. When other drivers send locations querying events on their upcoming routes, the traffic monitoring server returns results to drivers, helping them better understand the road conditions.

Despite the great potential held by traffic monitoring applications, challenging issues remain to be addressed for such location-based services. It is not an efficient way to query all events by sending two endpoints to the server, especially when the route is long. The reason is that events such as traffic congestion have timeliness, and drivers are expected to retrieve the latest information on the route. More importantly, uploading queries to the server will expose driver's locations and other potential attributes, which breaches the privacy requirement.

Many categories of location privacy-preserving mechanisms are proposed for location-based services (LBSs). $k$-anonymity first introduced in [4] can conceal a user's location into a cloaking zone with at least $k$ users. As a result, attackers cannot tell which of the $k$ users is the real one. Since then, several schemes based on the $k$-anonymity concept have been presented [14,18]. However, the effectiveness of $k$-anonymity lacks mathematical proof and the choice of $k$ is always biased. Dummy location is another method to protect location privacy where a user sends both real and dummy locations to the server [2,12]. However, dummy location-based schemes neglect the background knowledge of the attacker, thus privacy protection is compromised. Differential privacy (DP) provides privacy guarantee with strict mathematical proof for data sharing [3]. Authors in [1] developed a DP-based mechanism called geo-indistinguishability which protects the exact locations of individuals. However, one drawback is that perturbed locations are not always reasonable. Although some recent works such as [17] use point of interests (POIs) to find realistic locations, the issue is still unsolved as obfuscated locations heavily rely on the number of POIs on the map. For the identity-preserving techniques, pseudonym is mostly used to disrupt users' consistency. Nevertheless, pseudonym-based schemes are often presented with mix-zone to provide location privacy [9,10]. Besides, authors in [16] utilized fake queries to prevent the reverse mapping from user identity to query contents for continuous LBS. In [7], authors proposed a time-obfuscated algorithm where queries are sent to the server in random order.

To address the aforementioned challenges, we propose an efficient privacy-preserving scheme for traffic monitoring services. The route is first segmented into multiple sub-routes. Then we propose privacy-preserving mechanisms for both location and identity of drivers. The main contributions of this paper are summarized as follows: (i) we propose a location privacy-preserving mechanism based on geo-indistinguishability with mathematical proofs; (ii) we formalize the identity linking attack where a server can infer the identities of drivers through time intervals of continuous queries. Two solutions are proposed to prevent such attack. To the best of our knowledge, this is the *first* paper formally presenting such attack; and (iii) we conduct experiments on the real-world dataset to showcase the efficiency of our proposed scheme.

The remainder of this paper is organized as follows: We introduce background information in Sect. 2. The proposed scheme is presented in Sect. 3. The evaluation is described in Sect. 4. Finally, Sect. 5 concludes the paper.

## 2 Background

### 2.1 Definitions

**Differential Privacy.** Differential Privacy provides statistical information from a dataset while protecting the privacy of each individual, which can be described formally below.

**Definition 1** *($\epsilon$-Differential Privacy [3]). A privacy mechanism $\mathcal{M}$ gives $\epsilon$-differential privacy if any datasets $D_1$ and $D_2$ contain at most one different record, and for any possible output $S \in Range(\mathcal{M})$:*

$$\Pr\left(\mathcal{M}\left(D_1\right) \in S\right) \leq e^{\epsilon} \times \Pr\left(\mathcal{M}\left(D_2\right) \in S\right) \tag{1}$$

Differential privacy has a property of parallel composition, which indicates if differential privacy is utilized on disjoint databases, the privacy budget is determined by the worst privacy guarantee.

**Theorem 1** *(Parallel Composition [8]). Let function $\mathcal{M}_i$ each provides $\epsilon_i$-differential privacy. Applying each function over a set of disjoint databases $D_i$, the sequence of $\mathcal{M}_i$ provides $max\{\epsilon_i\}$-differential privacy.*

**Geo-indistinguishability.** Geo-indistinguishability ensures privacy preservation for places that are geographically close. In particular, given a chosen radius $r$ and privacy level $l$, a user achieves $\epsilon$-geo-indistinguishability for $\epsilon = l/r$.

**Definition 2** *($\epsilon$-Geo-indistinguishability [1]). A privacy mechanism $K$ satisfies $\epsilon$-geo-indistinguishability for any locations $l_1$ and $l_2$:*

$$Pr\left(K\left(l_1\right) = z\right) \leq e^{\epsilon d\left(l_1, l_2\right)} \times Pr\left(K\left(l_2\right) = z\right), \tag{2}$$

where $K$ denotes an algorithm generating the new location $z$ with actual location $l_1$ and $l_2$. $d\left(l_1, l_2\right)$ is the distance between $l_1$ and $l_2$.

### 2.2 Models

The system model includes three entities: driver, map server, and traffic monitoring server. Drivers submit locations to the traffic monitoring server querying events and receive results from the server. It is assumed that drivers send queries with different pseudonyms to protect identity privacy. The map server provides the route search service and always returns a route or returns null if no route exists between two given endpoints. The traffic monitoring server collects events

**Table 1.** Commonly used symbols

| Symbol | Description |
|---|---|
| $\mathcal{L}$ | Route |
| $l_i$ | $i^{th}$ segmented route in $\mathcal{L}$ |
| $l_i^s$ | Starting point of $l_i$ |
| $l_i^d$ | Destination of $l_i$ |
| $l_i^{os}$ | Obfuscated location of $l_i^s$ |
| $l_i^{od}$ | Obfuscated location of $l_i^d$ |
| $\langle p_i, p_j \rangle$ | Path between locations $p_i$ and $p_j$ |
| $\mathbb{P}$ | Path set from the map server |

(*e.g.,* traffic accidents and road congestion) happening on the roads from entities (*e.g.,* vehicles, roadside units, and pedestrians) in the vehicular network and provides traffic information to the requested drivers. Note that the process of information collection is beyond the scope of this paper.

In this paper, we assume that the map server is honest, but the traffic monitoring server is an *honest-but-curious* service provider that returns correct results to drivers but attempts to analyze drivers' information from queries. For example, the server performs *Bayesian attack* [6,11] where it induces the real location $l$ from the obfuscated location $l^{'}$ with a probability of $Pr(l|l^{'})$.

## 3 Our Proposed Scheme

### 3.1 Overview

The main purpose of our scheme is to ensure that drivers effectively query traffic information on their routes while fulfilling privacy requirements. In the proposed scheme, the whole route is truncated into multiple segments. When a driver is on the $(i-1)^{th}$ route, it is always querying events on the next segmented $i^{th}$ route to retrieve the most recent traffic information. Figure 1 illustrates the overview of our proposed scheme for each segmented route. A driver generates a radius $r$ based on a privacy budget $\epsilon$ and multiple locations $\{p_1, p_2, \ldots, p_k\}$. Then it sends these locations to a map server, which returns paths to the driver. The driver splits the circle with $r$ into multiple grids, and generates obfuscated locations. Moreover, to protect identity privacy of the driver, the query time is elaborately calculated. Finally, the driver sends two obfuscated endpoints at the specific time to the traffic monitoring server. Table 1 shows commonly used symbols in the paper.

### 3.2 Location Privacy-Preserving Mechanism

Given a specific driver's route $\mathcal{L}$, due to the timeliness concern, the scheme first segments the whole route evenly into multiple routes $l_i$: $l_i \in \{l_1, l_2, \ldots, l_n\}$

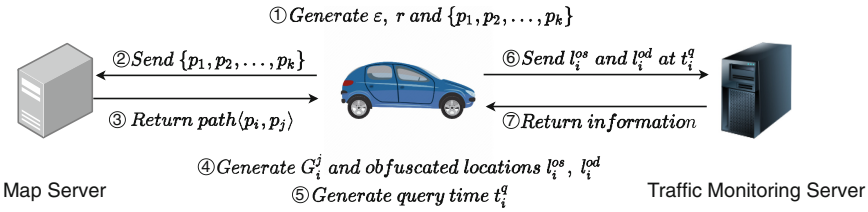



**Fig. 1.** Overview of our proposed scheme

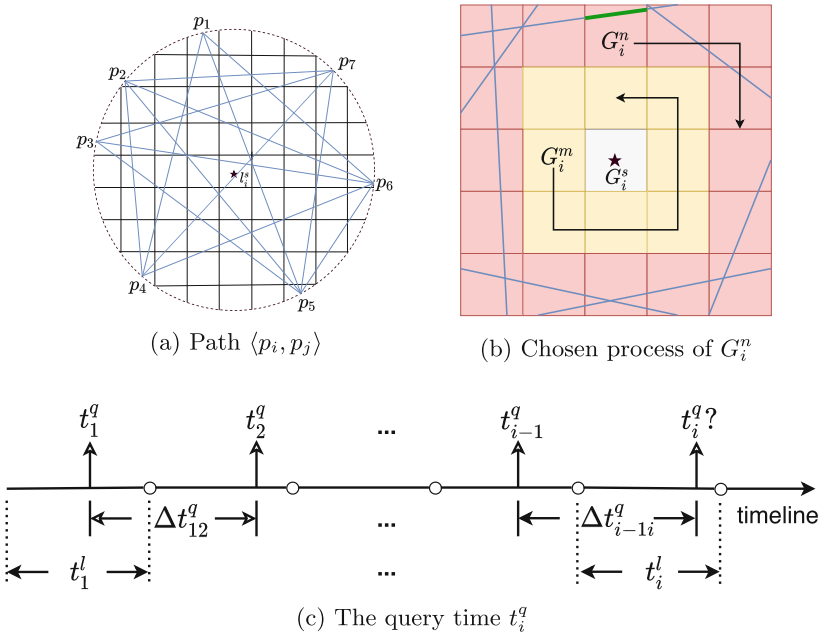
where $\sum_{i=0}^{n} |l_i| = |\mathcal{L}| \wedge |l_i| = |l_j| \wedge l_i \cap l_j = \emptyset, \forall i, j \in [1, n], i \neq j$. A driver always queries events on the route $l_{i+1}$ when he is on the route $l_i$. To achieve that, a driver needs to submit two endpoints of the current route (*i.e.,* $l_i^s$ and $l_i^d$) to the traffic monitoring server. Directly sending these endpoints will expose driver's locations to the server, thereby leaking the location privacy. Inspired by the geo-indistinguishability, we propose a privacy-preserving mechanism that guarantees location privacy by sending obfuscated locations. However, geo-indistinguishability has a significant drawback that obfuscated locations may not be geographically reasonable.

To tackle the issue, we propose a new algorithm that ensures that the obfuscated location is reasonable. Without losing the generality, we take the starting point $l_i^s$ as an example to illustrate the scheme in the rest of the paper. We first randomly select locations $\{p_1, p_2, \ldots, p_k\}$ on the circle with radius $r$ of $l_i^s$. The radius $r$ is generated by a given $\epsilon$ with Gamma distribution [1]. Then we send these locations to a map server requiring possible paths: $\langle p_i, p_j \rangle \in \mathbb{P}(p_i, p_j \in \{p_1, p_2, \ldots, p_k\})$. Be aware that the map server cannot identify $l_i^s$ since queries are sent with different pseudonyms and $\{p_1, p_2, \ldots, p_k\}$ are randomly chosen. Next, the circle is divided into grids $G_i^j$ and paths $\mathbb{P}$ are superimposed on the top of grids (illustrated in Fig. 2a). Particularly we denote the endpoint $l_i^s$ in the grid $G_i^s$. The driver maps the real location $l_i^s$ to the obfuscated location $l_i^{os}$ based on the Euclidean distance between $l_i^s$ and path $\langle p_i, p_j \rangle$ if the path exists in the grid $G_i^s$. The probability follows the exponential distribution and is calculated as follows:

$$Pr(\mathcal{M}(l_i^s) = l_i^{os}) = \frac{e^{-\frac{\epsilon}{2} d(l_i^s, l_i^{os})}}{\sum_{l_i^{os} \in S} \mathrm{e}^{-\frac{\epsilon}{2} d(l_i^s, l_i^{os})}}, \tag{3}$$

where $S$ denotes locations where $\langle p_i, p_j \rangle$ intersects with $G_i^s$. Equation (3) indicates that the nearest location is selected with the highest probability.

One possible case is that there is no path in the grid $G_i^s$ (*i.e.,* $\langle p_i, p_j \rangle \cap G_i^s = \emptyset$), as shown in Fig. 2b. We solve this by visiting other grids regarding to the distance between a grid and $G_i^s$. The neighboring grids of $G_i^s$ are first randomly visited in either a clockwise or anticlockwise manner. If there are still no paths in these grids, more outer grids will be visited until a path intersects with a grid. In the example of Fig. 2b, the green annotated path in the grid $G_i^n$ is chosen and




(a) Path $\langle p_i, p_j \rangle$

(b) Chosen process of $G_i^n$

(c) The query time $t_i^q$

**Fig. 2.** Demonstration of obfuscated location and query time. (Color figure online)

obfuscated location is then calculated based on Eq. (3). The location obfuscation process is also shown in Algorithm 1.

Similarly, the endpoint $l_i^d$ is obfuscated to $l_i^{od}$, and the driver sends both $l_i^{os}$ and $l_i^{od}$ to the server for $l_i$. By repeating the above process, other segmented routes are also obfuscated. In this case, the traffic monitoring server is unable to link these segmented routes to a complete one based on obfuscated locations, hence the location privacy is preserved. Our proposed location protection scheme provides $\epsilon$-geo-indistinguishability and is resistant to Bayesian attack with a bounded probability. We now theoretically prove them below.

**Theorem 2.** *Our proposed scheme provides $\epsilon$-geo-indistinguishability for the route.*

*Proof.* Let $\mathcal{M}_i^r$ be the location obfuscation mechanism $\mathcal{M}$ with radius $r$ for segmented route $l_i$. According to Theorem 1, our proposed scheme provides $\epsilon$-geo-indistinguishability for route $\mathcal{L}$ if $l_i$ guarantees $\epsilon$-geo-indistinguishability. Now we start to prove that $\mathcal{M}_i^r$ provides $\epsilon$-geo-indistinguishability for $l_i$. From Eq. (3), the ratio of $Pr(\mathcal{M}_i^r(l_i^s) = l_i^{os})$ to $Pr(\mathcal{M}_i^r(l_i^{s'}) = l_i^{os})$ is expressed as follows:

**Algorithm 1:** Location Obfuscation Mechanism

**Input:** Endpoint $l_i^s$ and $\epsilon$
**Output:** Obfuscated location $l_i^{os}$

Initialize $k \leftarrow 0$
$r \leftarrow GammaDis(\epsilon)$
Initialize locations $\{p_1, p_2, \ldots, p_k\}$
$\mathbb{P} \leftarrow RetrivePath(p_1, p_2, \ldots, p_k)$
$\{G_i^1, G_i^2, \ldots, G_i^n\} \leftarrow SplitCircle(r)$
**foreach** $G_i^m$ *in* $\{G_i^1, G_i^2, \ldots, G_i^n\} \wedge Dist(G_i^m, G_i^s) == k$ **do**
  **foreach** $\langle p_i, p_j \rangle \in \mathbb{P}$ **do**
    **if** $\langle p_i, p_j \rangle \cap G_i^m = \emptyset$ **then**
      *continue*;
    **end**
    **else**
      $S \leftarrow \langle p_i, p_j \rangle \cap G_i^m$;
      $l_i^{os} \leftarrow GetLocation(l_i^s, \epsilon, S)$; `// Using Eq. (3) to calculate the obfuscated location`
      return $l_i^{os}$;
    **end**
  **end**
  $++k$;
**end**
**return** *null*;

$$\frac{Pr(\mathcal{M}_i^r(l_i^s) = l_s^{os})}{Pr(\mathcal{M}_i^r(l_i^{s'}) = l_i^{os})} = \frac{\sum_{l_i^{os} \in S} \mathrm{e}^{-\frac{\epsilon}{2} d(l_i^{s'}, l_i^{os})} \cdot e^{-\frac{\epsilon}{2} d(l_i^s, l_i^{os})}}{\sum_{l_i^{os} \in S} \mathrm{e}^{-\frac{\epsilon}{2} d(l_i^s, l_i^{os})} \cdot e^{-\frac{\epsilon}{2} d(l_i^{s'}, l_i^{os})}}$$
$$= \frac{\sum_{l_i^{os} \in S} \mathrm{e}^{-\frac{\epsilon}{2} d(l_i^{s'}, l_i^{os})}}{\sum_{l_i^{os} \in S} \mathrm{e}^{-\frac{\epsilon}{2} d(l_i^s, l_i^{os})}} \cdot e^{\frac{\epsilon}{2}(d(l_i^{s'}, l_i^{os}) - d(l_i^s, l_i^{os}))} \tag{4}$$

Due to the triangle inequality, $\forall l_i^s, l_i^{s'}$, we show that the following inequalities hold:

$$d(l_i^{s'}, l_i^{os}) - d(l_i^s, l_i^{os}) \leq d(l_i^s, l_i^{s'}) \tag{5}$$

$$e^{-\frac{\epsilon}{2} d(l_i^{s'}, l_i^{os})} \leq e^{-\frac{\epsilon}{2}(d(l_i^s, l_i^{os}) - d(l_i^s, l_i^{s'}))} \tag{6}$$

Using Eq. (5) and Eq. (6), we have the following inequality:

$$\sum_{l_i^{os} \in S} \mathrm{e}^{-\frac{\epsilon}{2} d(l_i^{s'}, l_i^{os})} - \mathrm{e}^{\frac{\epsilon}{2} d(l_i^s, l_i^{s'})} \cdot \sum_{l_i^{os} \in S} \mathrm{e}^{-\frac{\epsilon}{2} d(l_i^s, l_i^{os})}$$
$$= \sum_{l_i^{os} \in S} (\mathrm{e}^{-\frac{\epsilon}{2} d(l_i^{s'}, l_i^{os})} - e^{-\frac{\epsilon}{2}(d(l_i^s, l_i^{os}) - d(l_i^s, l_i^{s'}))}) \leq 0 \tag{7}$$

Therefore, we obtain:

$$\frac{\sum_{l_i^{os} \in S} \mathrm{e}^{-\frac{\epsilon}{2} d(l_i^{s'}, l_i^{os})}}{\sum_{l_i^{os} \in S} \mathrm{e}^{-\frac{\epsilon}{2} d(l_i^{s}, l_i^{os})}} \leq \mathrm{e}^{\frac{\epsilon}{2} d(l_i^{s}, l_i^{s'})} \tag{8}$$

Finally, the following inequality is derived:

$$\frac{Pr(\mathcal{M}_i^r(l_i^s) = l_i^{os})}{Pr(\mathcal{M}_i^r(l_i^{s'}) = l_i^{os})} \leq \mathrm{e}^{\frac{\epsilon}{2} d(l_i^{s}, l_i^{s'})} \cdot \mathrm{e}^{\frac{\epsilon}{2} d(l_i^{s}, l_i^{s'})} = \mathrm{e}^{\epsilon d(l_i^{s}, l_i^{s'})} \tag{9}$$

The proof shows that $\mathcal{M}_i^r$ provides $\epsilon$-geo-indistinguishability. Therefore, our scheme satisfying $\epsilon$-geo-indistinguishability has been proved.

**Theorem 3.** *Our proposed scheme is resistant to the Bayesian attack with a bounded probability.*

*Proof.* Similarly, the whole route is resistant to the Bayesian attack if every segmented one does. We now prove the mechanism resistant to the Bayesian attack for $l_i$. According to the attack model, the adversary tries to estimate the actual location $\hat{l}$ by maximizing the posterior probability $Pr(l_i^s | l_i^{os})$ as follows:

$$\hat{l} = arg \max_{l_i^s \in \Theta} Pr(l_i^s | l_i^{os}) \tag{10}$$

where $\Theta$ denotes the location set. Thus, the probability is calculated:

$$\begin{aligned} Pr(l_i^s \mid l_i^{os}) &= \frac{\varphi(l_i^s) Pr(l_i^{os} \mid l_i^s)}{\sum_{l_i^{s'} \in \Theta} \varphi(l_i^{s'}) Pr(l_i^{os} \mid l_i^{s'}) + \sum_{l_i^{s'} \notin \Theta} \varphi(l_i^{s'}) Pr(l_i^{os} \mid l_i^{s'})} \\ &\leq \frac{\varphi(l_i^s) Pr(l_i^{os} \mid l_i^s)}{\sum_{l_i^{s'} \in \Theta} \varphi(l_i^{s'}) Pr(l_i^{os} \mid l_i^{s'})} \\ &= \frac{\varphi(l_i^s)}{\sum_{l_i^{s'} \in \Theta} \varphi(l_i^{s'}) Pr(l_i^{os} \mid l_i^{s'}) / Pr(l_i^{os} \mid l_i^s)} \\ &\leq e^{\epsilon d(l_i^s, l_i^{s'})} \frac{\varphi(l_i^s)}{\sum_{l_i^{s'} \in \Theta} \varphi(l_i^{s'})} \end{aligned}$$

where $\varphi(\cdot)$ denotes the prior information of an actual location. The proof indicates that the probability of successfully capturing the real location is limited by the $\epsilon$-geo-indistinguishability no matter what prior knowledge the adversary has.

### 3.3 Identity Privacy-Preserving Mechanism

As the route is evenly segmented, query intervals are almost the same. An attacker can infer whether queries are sent from the same identity by analyzing the time intervals between adjacent queries. First, we give the formal definition of identity linking attack.

**Definition 3** *(Identity Linking Attack). Given a sequence of information queries* $\{q_1, q_2, \ldots, q_k\}$ *sent at corresponding timing* $\{t_1^q, t_2^q, \ldots, t_k^q\}$ *with different identities* $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_k\}$*. When time intervals fulfill:* $(t_{i+2} - t_{i+1}) - (t_{i+1} - t_i) \leq \delta, \forall i \in [1, k-2] \wedge \delta \geq 0$*, an attacker can infer identities* $\mathcal{I}_{i+2} = \mathcal{I}_{i+1} = \mathcal{I}_i$*.*

In this attack, when a driver sends queries at two adjacent intervals less than a threshold $\delta$, a server can link queries to the same driver, thus the identity privacy is leaked even though the pseudonym technique is used.

To achieve identity unlinkability, we propose two solutions: random-based and statistical solutions. random-based solution is a simple one where the query time is randomly selected in the route $l_{i-1}$ when a driver queries events for the next route $l_i$. Intuitively, the server is difficult to link identity when time intervals are different. Therefore, the statistic solution maximizes the variance of time intervals before sending a query, which is demonstrated in Fig. 2c. Specifically, a driver records the time $t_k^q$ and the time cost $t_k^l$ of route $l_k$. Then it calculates the time intervals $\mathbb{T} = \{\Delta t_{12}^q, \Delta t_{23}^q, \ldots, \Delta t_{i-2i-1}^q\} (\Delta t_{ij}^q = t_j^q - t_i^q)$. Before sending the $i^{th}$ query, the driver calculates the query sent time $t_i^q$ which should fulfill:

$$\begin{aligned} &\max_{t_i^q} \ \Delta t_{i-1i}^q - \overline{\Delta t^q}, \\ &s.t. \ t_i^q \geq \sum t_{i-1}^l, \\ &\quad t_i^q \leq \sum t_{i-1}^l + \overline{t^l}, \\ &\quad t_i^q > 0, i = 1, 2, \ldots, k. \end{aligned} \tag{11}$$

where $\overline{\Delta t^q}$ and $\overline{t^l}$ denote the average time interval in $\mathbb{T}$ and the average time of $\{t_1^l, t_2^l, \ldots, t_i^l\}$, respectively. The constraints guarantee that $t_i^q$ is always bounded.

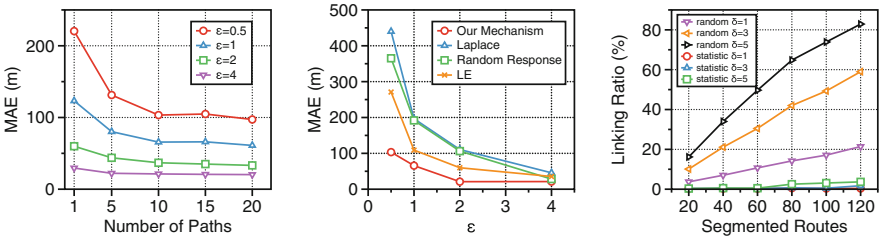
## 4 Performance Evaluation

We evaluate the scheme in two aspects: privacy-preserving mechanisms and costs. Since the scheme applies geo-indistinguishability to protect location privacy, it inevitably reduces the service usability for route query. We use the *mean absolute error* (MAE) to quantify service usability [13]. Moreover, identity privacy is evaluated by the linking ratio where the number of successful linking is divided by the total number of time intervals. The low ratio indicates the high level of identity privacy. The experiments of mechanism evaluation are conducted on the real-world dataset GeoLife[1]. We evaluate the costs using Baidu Map API on a Windows 11 desktop with 4.20 GHz AMD Ryzen 5 processor.

### 4.1 Results of Privacy-Preserving Mechanisms

Figure 3a illustrates results of MAE with $\epsilon$ varying in $\{0.5, 1, 2, 4\}$ and number of paths in $\{1, 5, 10, 15, 20\}$. It is figured that the MAE decreases with the increase

[1] https://www.microsoft.com/en-us/download/details.aspx?id=52367.



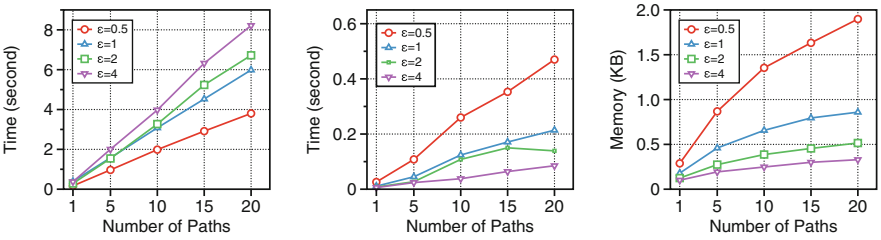

(a) MAE (b) Comparison (c) Identity Privacy

**Fig. 3.** Results of privacy-Preserving mechanisms

of number of paths. However, continuous growing of number of paths cannot further decrease the MAE. For example, the MAE decreases from 220 m to 100 m when paths change from 1 to 10 with $\epsilon$ 0.5, and fluctuates around 100 m. It indicates that simply increasing the path number cannot effectively make the obfuscated location closer to the real location due to the location privacy-preserving mechanism. Besides, it is also observed that the MAE increases with the decrease of $\epsilon$, which is in accordance with the definition of differential privacy.

To further evaluate the usability of the mechanism, we compare it with other differential privacy mechanisms including Laplace, random response, and linear equations (LE) [5]. The number of paths is set to 10 in the simulation. Results in Fig. 3b show that our proposed mechanism achieves the lowest MAE compared with others especially when $\epsilon$ are 0.5, 1, and 2. When $\epsilon$ is 0.5, in our mechanism the obfuscated location is 100 m away from the real location, while the distances are over 250 m for LE and 450 for Laplace. This means our mechanism causes the smallest distance deviation for the same level of differential privacy, hence providing better service usability than other mechanisms.

Figure 3c presents the results of identity privacy. We compare both the random-based solution and statistic solution described in Sect. 3.3. The thresholds $\delta$ are set to 1, 3, and 5 s, and the number of segmented routes varies from 20 to 120. Results show that the random-based solution performs much worse than the statistic one especially when the threshold and the number of routes increase. The reason is that the query timing in random-based solution is completely random in each segmented route and thus two time intervals have a very high probability to be similar. In this case, the attacker can easily perform identity linking attack. For example, when the threshold is set to 5 s and routes are 120, the linking ratio reaches over 80% for random-based solution, meaning that 80% of queries are sent from the same identity. In contrast, the linking ratios in statistic solution are always less than 10%, which indicates its effectiveness for protecting identity privacy.




(a) Communication Cost (b) Computation Cost (c) Memory Cost

**Fig. 4.** Results of costs

### 4.2 Results of Costs

In this section, we evaluate time cost and memory cost in the scheme. The number of paths varies from 1 to 20. Communication cost refers to the time delay between a driver sending request and receiving paths from a map server. Computation cost is the time delay of calculating the obfuscated location.

In Fig. 4a, it is figured that the time cost increases with the number of paths as the map server spends more time calculating paths. Besides, the large $\epsilon$ causes more time cost. This is due to the fact that searching reasonable and not duplicated paths in a small area is difficult. The results also show that the communication time is less than 10 s with 20 paths when $\epsilon$ is 4, meaning that such quantity of delay is acceptable in practice. Note that the time cost may increase when a driver requires a map server to return more paths. However, the MAE will not further decrease as we have explained before.

On the other hand, as shown in Fig. 4b, the computation delay is always less than 1 second, which is negligible compared with the communication delay. Moreover, in Fig. 4c, the memory cost is less than 2 KB for 20 paths, showing that such cost is also trivial. Results indicate that the performance of our proposed scheme is not severely affected by introducing such level of overhead.

## 5 Conclusion

This paper introduces a privacy-preserving scheme for traffic monitoring services in vehicular networks. We emphasize solving privacy issues when drivers send queries to the cloud server for traffic information. Both location privacy and identity privacy are considered. In the scheme, the real location is perturbed to a reasonable location with $\epsilon$-geo-indistinguishability guarantee. We also propose a new type of attack called identity linking attack in the continuous-query environment. Two possible solutions are presented to prevent such attack. We implement the scheme on the real dataset, and results show the efficiency and usability of our proposed scheme.

## References


1. Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: Differential privacy for location-based systems. In: ACM Conference on Computer and Communications Security, pp. 901–914 (2013)
2. Chen, S., Shen, H.: Semantic-aware dummy selection for location privacy preservation. In: 2016 IEEE Trustcom/BigDataSE/ISPA, pp. 752–759 (2016)
3. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79228-4_1
4. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: International Conference on Mobile Systems, Applications and Services, pp. 31–42 (2003)
5. Gu, X., Li, M., Cao, Y., Xiong, L.: Supporting both range queries and frequency estimation with local differential privacy. In: IEEE Conference on Communications and Network Security, pp. 124–132 (2019)
6. Gursoy, M.E., Tamersoy, A., Truex, S., Wei, W., Liu, L.: Secure and utility-aware data collection with condensed local differential privacy. IEEE Trans. Depend. Secure Comput. **18**(5), 2365–2378 (2021)
7. Hwang, R.H., Hsueh, Y.L., Chung, H.W.: A novel time-obfuscated algorithm for trajectory privacy protection. IEEE Trans. Serv. Comput. **7**(2), 126–139 (2014)
8. McSherry, F.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. Commun. ACM **53**(9), 89–97 (2010)
9. Memon, I., Ali, Q., Zubedi, A., Mangi, F.: DPMM: dynamic pseudonym-based multiple mix-zones generation for mobile traveler. Multimedia Tools Appl. **76**, 1–30 (2017)
10. Memon, I., Memon, H., Arain, Q.A.: Pseudonym changing strategy with mix zones based authentication protocol for location privacy in road networks. Wirel. Pers. Commun. **116**, 3309–3329 (2021)
11. Niu, B., Chen, Y., Wang, Z., Li, F., Wang, B., Li, H.: Eclipse: preserving differential location privacy against long-term observation attacks. IEEE Trans. Mob. Comput. **21**(1), 125–138 (2022)
12. Sun, G., et al.: Efficient location privacy algorithm for internet of things (IoT) services and applications. J. Netw. Comput. Appl. **89**, 3–13 (2017)
13. Wang, L., Zhang, D., Yang, D., Lim, B.Y., Han, X., Ma, X.: Sparse mobile crowdsensing with differential and distortion location privacy. IEEE Trans. Inf. Forensics Secur. **15**, 2735–2749 (2020)
14. Xing, L., Jia, X., Gao, J., Wu, H.: A location privacy protection algorithm based on double k-anonymity in the social internet of vehicles. IEEE Commun. Lett. **25**(10), 3199–3203 (2021)
15. Xu, W., Zhou, H., Cheng, N., Lyu, F., Shi, W., Chen, J., Shen, X.: Internet of vehicles in big data era. IEEE/CAA J. Autom. Sin. **5**(1), 19–35 (2018)
16. Ye, A., Li, Y., Xu, L.: A novel location privacy-preserving scheme based on l-queries for continuous LBS. Comput. Commun. **98**, 1–10 (2017)
17. Zhang, P., Hu, C., Chen, D., Li, H., Li, Q.: ShiftRoute: achieving location privacy for map services on smartphones. IEEE Trans. Veh. Technol. **67**(5), 4527–4538 (2018)
18. Zhang, S., Li, X., Tan, Z., Peng, T., Wang, G.: A caching and spatial k-anonymity driven privacy enhancement scheme in continuous location-based services. Future Gen. Comput. Syst. **94**, 40–50 (2019)

# Skin Lesion Segmentation via Intensive Atrous Spatial Transformer

Xiuli Liu, Wanshu Fan(✉), and Dongsheng Zhou(✉)

National and Local Joint Engineering Laboratory of Computer Aided Design, School of Software Engineering, Dalian University, Dalian 116622, LiaoNing, China
fan921amber@163.com, zhouds@dlu.edu.cn

**Abstract.** Skin melanoma is one of the most malignant tumors. In recent years, its incidence rate and mortality showed a high growth trend. Early detection and segmentation of skin lesions are vital in timely diagnosis and treatment. As the low contrast of lesion regions and high similarity in terms of appearance, skin lesion segmentation still remains a challenging work. Most of the segmentation methods use single-scale feature fusion, leading to the blur effect on the boundary. In this paper, we propose a new segmentation framework named Intensive Atrous Spatial Transformer Network (IASTrans-Net), which is based on a core module Intensive Atrous Spatial Pyramid Pooling (IASPP). The introduced IASPP can obtain valid features by using multi-scale feature fusion and channel attention. On the one hand, we employ atrous convolution with different dilation rates for multi-scale information extraction, ensuring that the effective information of each channel is obtained. On the other hand, channel attention is used to screen features, which can enable the network to effectively identify targets without increasing the training complexity. The experimental results show that the proposed IASTrans-Net has achieved good results in ISIC2017 and ISIC2018 datasets, surpassing most of the current mainstream methods.

**Keywords:** Transformer · Atrous convolution · Multi-scale fusion · Channel attention

## 1 Introduction

According to the statistics of the World Health Organization, about 100000 cases of skin melanoma and skin cancer are diagnosed every year [18]. Although the mortality rate is fairly significant, early detection and segmentation of melanoma are able to improve the survival rate to over 95% [20]. This underlines the importance for timely diagnosis and treatment of melanoma. Computer-aided diagnostic system is thus essential for detecting and segmenting the skin lesion automatically and accurately, and the segmentation results can help to identify regions-of-interest for skin lesion assessment [4]. However, as the skin lesions have the characteristic of low contrast, high similarity between appearance and


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 15–26, 2022.
https://doi.org/10.1007/978-3-031-19208-1_2

healthy skin, and they may also affected by the influence of hair occlusion and acquisition equipment, the automatic segmentation of skin lesions in dermoscopic images is a very challenging work [8]. Some researchers deal with skin lesion segmentation by utilizing classical image processing technologies such as thresholding, edge based or region based methods [17]. Nevertheless, these aforementioned methods mainly rely on relatively simple hand-crafted features, which do not always reflect the key information of the image [10].

In recent years, deep learning methods have made remarkable progress in the filed of semantic segmentation. Compared with traditional hand-crafted feature methods, deep learning based methods can avoid the hand-crafting process of features and have the capability to capture more significant features from the whole image. These methods focus on designing various deep networks to tackle skin lesion segmentation. Full Convolution Network (FCN) [15] is a pioneering work for image segmentation. It converts the full connection layer of traditional CNN into convolution layer one by one, and solves the problem of low convolution calculation efficiency of each pixel block. U-Net [16] is another widely used encoder-decoder network architecture for many segmentation tasks. PSPNet [24], DeepLab [6], Mask RCNN [11], Non-local U-Nets [22], and CRF-RNN [25] further improves the performance of CNN in the field of image segmentation.

In this paper, we propose an IASTrans-Net. The proposed IASTrans-Net focuses on multi-scale feature fusion and uses different dilation rates to capture multi scale information, which effectively gains different receptive fields and retains the effective information in the image to the greatest extent. The attention module in our designed IASPP module can screen the fused features, adaptively fuse the feature points of each skin lesion, and realize the effective segmentation of melanoma skin lesion images. Experimental results compared with some advanced methods demonstrate the effectiveness of the proposed method.

In general, our main contributions of this paper are as follows:

- A multi-scale feature fusion attention network IASTrans-Net is proposed to solve skin lesion segmentation. The network uses context information to extract features in the coding layer, which improves the representation ability of the features.
- In this paper, we match the atrous convolution with different dilation rates for the feature maps of different scales. For each feature map of specific scale, the convolution layers of multiple dilation rates can fully mine the pathological skin features under different dilation rates, and obtain a larger receptive field without losing resolution, so as to obtain effective information to the greatest extent.
- For the obtained feature information, we utilize a attention module for screening. The multi receptive field features obtained at different levels are complementary to the feature information of local cross-channel interaction, the high-level and low-level features are systematically integrated. Capturing different granularity information and effectively identifying different types of features are very important for segmentation tasks.
- Quantitative and qualitative experiments on ISIC2017 and ISIC2018 datasets show that the proposed method is superior to some advanced methods.

## 2 Related Works

Most of the existing image segmentation methods can be roughly divided into two categories: hand-crafted feature based methods and deep learning based methods.

### 2.1 Hand-Crafted Feature Based Methods

Most early image segmentation algorithms attempt to explore hand-crafted features to segment the lesions. Wong et al. [23] proposed a most intuitive iterative random region merging method, which segmented the region corresponding to skin lesions from the macro image, and introduced the region merging likelihood function based on region statistics to determine the region merging in a random manner. However, the steps of this method are cumbersome, and some processes need to be repeated. Therefore, Garnavi et al. [9] proposed an automatic segmentation method based on color space analysis and histogram threshold clustering, which can determine the best color channel for skin lesion segmentation. However, this method is sensitive to noise and has high computational complexity when applied to RGB image multi threshold segmentation. Subsequently, Liu et al. [14] proposed a algorithm of light modeling and pigment mass recognition based on decomposition and weighted polynomial curve fitting, which is a adaptive method for extracting melanin of human skin. These methods rely on hand-crafted features and cannot capture high-level feature information.

### 2.2 Deep Learning Based Methods

Recently, the amazing success of deep learning provides new ideas for image segmentation. Generally, among different deep learning based methods, FCN [15] and its extension U-Net [16] are widely utilized in semantic image segmentation. Vivek et al. [19] proposed an accurate skin segmentation model based on improved condition Generation Countermeasure Network (cGAN) to solve the problems of low contrast and rough boundary of skin lesions. Recently, some researchers consider that CNN is inefficient in capturing global image context information, and Transformer can make up for this shortcoming. With the emergence of ViT model [13], Transformer has developed rapidly in the field of computer vision, and has been well applied in the fields of human pose estimation, semantic segmentation, instance segmentation and medical image segmentation. However, the pure Transformer network has a high demand for graphics cards and is not suitable for small tasks. Chen et al. [5] proposed a TransUNet, which is the first attempt of using Transformer to solve medical image segmentation problem. This network transforms the image into a sequence and encodes the global information, but ignores the internal structure features of each patch at the pixel level.

While most deep learning based methods focus on designing effective depth CNN architecture, leading to losing depth low resolution features and fine-grained features, so as to get some rough segmentation boundary templates. To overcome this deficiency, we propose an IASTrans-Net in this paper.

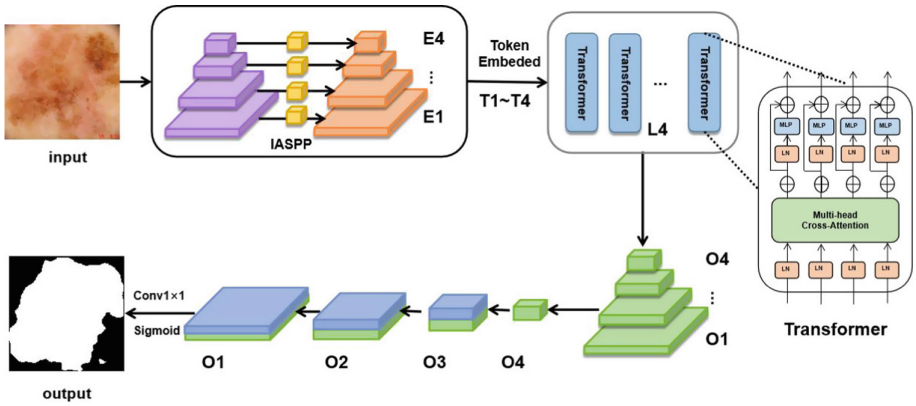



**Fig. 1.** Overall Network Framework. IASTrans-Net first extracts the features of different scales, and then processes them through IASPP module and Transformer module to obtain a set of feature maps O1–O4. In turn, each feature map is up sampled and reconstructed by convolution layer. Finally, $1 \times 1$ convolution and sigmoid operation are used for the obtained feature map to get segmentation maps.

## 3 The Proposed Method

### 3.1 Overall Framework

For image segmentation task, the key to accurately segment the lesion edge is effectively fuse the multi-scale context information and make full use of the long-range dependence of features. The IASTrans-Net proposed in this paper uses atrous convolution to fuse the features of different layers and adequately consider the long-range dependence of transformer, so as to better integrate the features of the encoder and reduce the semantic gap. The overall network framework is shown in Fig. 1.

Firstly, the input image extracts the depth features through ResNet-34, and sends them to IASPP module (We will give detailed analysis on the module of IASPP in Sect. 3.2). The module captures the effective features on each scale through atrous convolution and channel attention. Specifically, the corresponding feature maps is: $E_i \in R^{\frac{HW}{i^2} \times C^i}, (i = 1, 2, 3, 4)$. $H$ denotes for the height, $W$ denotes the width and $C$ denotes the number of channels.

Secondly, the feature map is tokenized to have location information, and then send token into the Transformer. It uses the transformer structure in UCTransNet [21] and it includes multi-head cross-attention module and multi-layer perceptron with residual structure. The channel relationship and feature dependence are obtained by Eq.(1) and Eq. (2).

The multi-head cross-attention module contains five inputs, including four tokens $T_i$ as queries and a connected token $T_\Sigma$ as key and value:

$$Attention(Q_i, K, V) = soft\max(\frac{Q_i^T K}{\sqrt{C_\Sigma}})V^T, \tag{1}$$

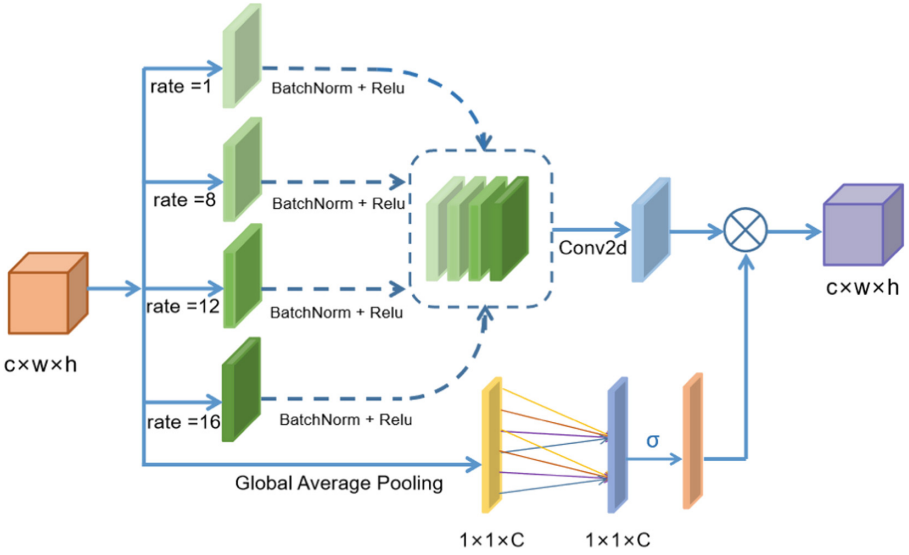



**Fig. 2.** IASPP module. The dilation rate is used by the feature of the second layer. For adapting the features with different scales, different dilation rates are used for each layer.

$$Q_i = T_i W_{Q_i}, K = T_\Sigma W_K, V = T_\Sigma W_V, \tag{2}$$

where $Q_i \in R^{C_i \times d}, K \in R^{C_\Sigma \times d}, V \in R^{C_\Sigma \times d}$ are produced by a cross-attention mechanism in Transformer. $W_{Q_i} \in R^{C_i \times d}, W_k \in R^{C_\Sigma \times d}, W_V \in R^{C_\Sigma \times d}$ are weights for different inputs, $d$ is the sequence length, $C_i (i = 1, 2, 3, 4)$ are the channel size of four skip connection layers. In the process of implementation, $C_1 = 64, C_2 = 128, C_3 = 256, C_4 = 512$. The output of multi-head cross-attention is the result of applying simple MLP and residual operator to calculate the average value of attention.

Thirdly, in order to better integrate the features of different scales of Transformer, we obtain a set of feature maps $O_1$–$O_4$. In turn, $O_4$ is up sampled and convolution layer reconstructed, spliced with $O_3$, then the obtained feature map is spliced with $O_2$.

Finally, $1 \times 1$ convolution and sigmoid operation are used for the obtained feature map to get segmentation maps.

### 3.2 IASPP Module

Considering the significant of local and global features for accurately segmenting of lesion boundary, we design an attention module that can make full use of context called IASPP. The proposed IASPP module is shown in Fig. 2.

The attention module is mainly composed of two parts. The first part uses atrous convolution to extract information, and the second part mainly uses channel attention to screen effective information.

In the first part, firstly, the feature map of different scales and four atrous convolutions with different dilation rates are used to extract the features of the feature map, so that the network can obtain different receptive fields without losing the resolution and capture information at different scales. For each position $i$ on the output $y$ and a filter $w$, we use the atrous convolution on the input feature map $x$:

$$y[i] = \sum_k x[i + r \cdot k]w[k], \tag{3}$$

the operation of Eq. (3) is to perform the atrous convolution with different dilation rates for each input feature map $x$, where $r$ represents the dilation rate using atrous convolution. Equivalent to the input feature map $x$, the convolution kernel using $3 \times 3$ in the convolution process is filled by inserting $r$-1 zeros between every two values of the convolution kernel. Generally, ordinary convolution is equivalent to the case of $r = 1$, i.e., no filling value. When atrous convolution is used, the size of convolution kernel is controlled by changing the value of $r$ to obtain different receptive fields without reducing the resolution.

Secondly, we process the data with a batchnorm and a relu layer. The reason to obtain multi-scale feature maps for processing lies in that it can extract multiple resolution information as features.

Finally, the extracted features are concatenated to enrich the extracted feature information and improve the segmentation performance.

In the second part, an ultra lightweight channel attention operation is carried out in parallel. The attention is composed of a global average pooling, an adaptive kernel size one-dimensional convolution and sigmoid activation function. Through this operation, we can get a feature map that captures local cross channel interaction. Then the feature map is used to filter out the invalid features of other channels. In this way, the fuzziness of features is eliminated. The multi receptive field features obtained at multiple levels are complementary to the feature information of local cross-channel interaction to realize multi-scale context aggregation. Compared with group convolution and depth separable convolution, this module achieves good results in the case of low complexity.

The matrix $\omega_{\mathrm{k}}$ used by this module is as follows:

$$\begin{bmatrix} \omega^{1,1} & \cdots & \omega^{1,k} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & \omega^{2,2} & \cdots & \omega^{2,k+1} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \omega^{C,C-k+1} & \cdots & \omega^{C,C} \end{bmatrix} \tag{4}$$

$\omega_k$ consists of $k \times C$ parameters, which avoids the complete independence between different channels and takes into account the interaction between different channels.

$$\omega_i = \sigma\left(\sum_{j=1}^{k} w_i^j y_i^j\right), y_i^j \in \Omega_i^k, \tag{5}$$

the calculation of $y_i$ weight only considers the interaction between $y_i$ and its $k$ nearest neighbors, where $\Omega_i^k$ represents the set of $k$ adjacent channels of $y_i$.

### 3.3 Loss Function

In this paper, we use cross entropy as the loss function, it is mainly used to measure the difference information between the probability distribution of the predicted value and the target value. The cross entropy loss is calculated as :

$$L_{CE} = \frac{1}{N}\sum_{i}^{N}[y_i \log \widehat{y_i} + (1 - y_i)\log(1 - \widehat{y_i})]. \tag{6}$$

where $N$ represents the number of pictures processed in each batch and $i$ denotes the spatial location of a pixel, $y_i$ is the label value and $\widehat{y_i}$ is the predicted value of the model.

## 4 Experiments

### 4.1 Experiment Settings

**Datasets**. In order to evaluate the effectiveness of the proposed IASTrans-Net, we have used two challenging datasets: ISIC2017 [7] and ISIC2018 [7]. ISIC2017 dataset contains 2000 training set dermatoscope images, 150 verification set images and 600 test set images. ISIC2018 dataset contains 2594 training sets and 1000 test sets of dermoscopic images.

**Evaluation Criterions.** Jaccard(JA), Dice(DI) and Accuracy(AC) are commonly used metrics to measure the similarity between the prediction and the ground truth for image segmentation. We adopt JA, DI and AC as our measurement criteria.

**Implementing Details.** We first resize all images to $256 \times 256$. For model training, we use Adam to optimize our network, and the batch size is chosen as 4. To improve the stability of training, the initial learning rate is set to $1e \times 10^{-3}$, and the learning rate for every 10 epochs is reduced by ten times, $1e \times 10^{-4}$, we train the model to 50 epochs. In the training process, the 26th epoch has the best training effect.

### 4.2 Experimental Results

Quantitative comparisons between our method and other advanced methods are shown in Table 1. On ISIC2017 dataset, we compare our method with six advanced methods, including U-Net [16], SwinUNet [3], UCTransNet [21], TransUNet [5], SkinNet [20], FrCN [1]. It can be seen that our method generates the results with higher DI, JA and AC among all the compared methods.

On ISIC2018 dataset, we compare our method with six advanced methods, including U-Net [16], UCTransNet [21], SwinUNet [3], TransUNet [5], MCGU-Net [2] and DoubleU-Net [12]. Our method also achieves the highest scores of the evaluative criteria on this dataset. It illustrates that the adaptive multi-scale context guided segmentation is very useful for multi-layer space in the process of deep learning. The segmentation diagrams of some advanced methods are shown in Fig. 3.

**Table 1.** Compared with some mainstream methods on ISIC2017 and ISIC2018 datasets, bold numbers represent the best performance of each column.

| Datasets | Methods | DI | JA | AC |
|---|---|---|---|---|
| ISIC-2017 | U-Net | 77.03 | 67.15 | 90.52 |
| | SwinUNet | 77.29 | 66.51 | 91.21 |
| | UCTransNet | 78.90 | 69.35 | 91.29 |
| | TransUNet | 83.58 | 74.33 | 92.84 |
| | SkinNet | 85.50 | 76.70 | 93.20 |
| | FrCN | 87.08 | 77.11 | 94.03 |
| | Ours | **87.89** | **80.19** | **94.59** |
| ISIC-2018 | U-Net | 83.62 | 75.17 | 91.75 |
| | UCTransNet | 86.41 | 78.60 | 93.13 |
| | SwinUNet | 86.02 | 78.21 | 92.96 |
| | TransUNet | 88.32 | 81.25 | 93.67 |
| | MCGU-Net | 89.50 | – | 95.5 |
| | DoubleU-Net | 89.62 | – | – |
| | Ours | **90.18** | **83.79** | **94.80** |

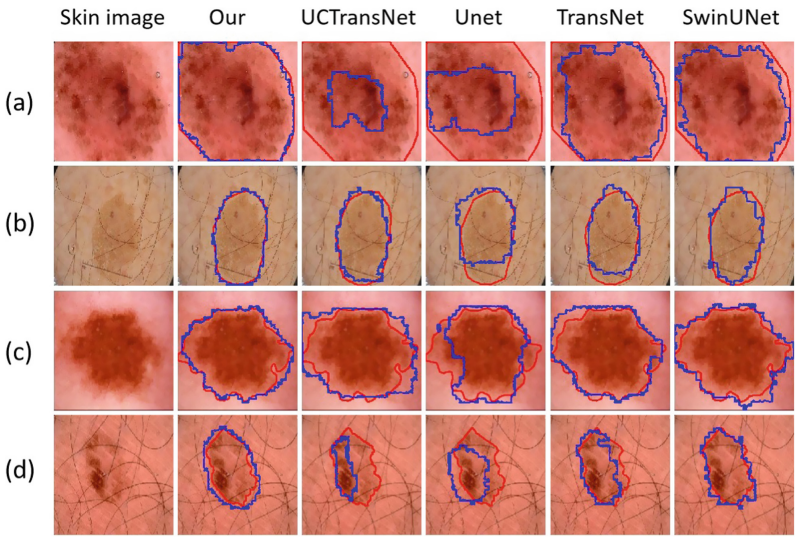



**Fig. 3.** Segmentation results on ISIC2017 and ISIC2018 datasets, where (a), (b) are the results on the ISIC2017 dataset, (c), (d) are the results on the ISIC2018 dataset. The red and blue outlines represent ground truth and segmentation results, respectively.

### 4.3 Analysis

**Ablation Study**. It is significant to explore the effectiveness of our proposed module in context information processing at different levels of the network. To this end, we carry out an ablation study on five different structures: the IASPP module is embedded in the first layer to the fourth layer and all layers respectively. The segmentation results of different layers are presented in Fig. 4.

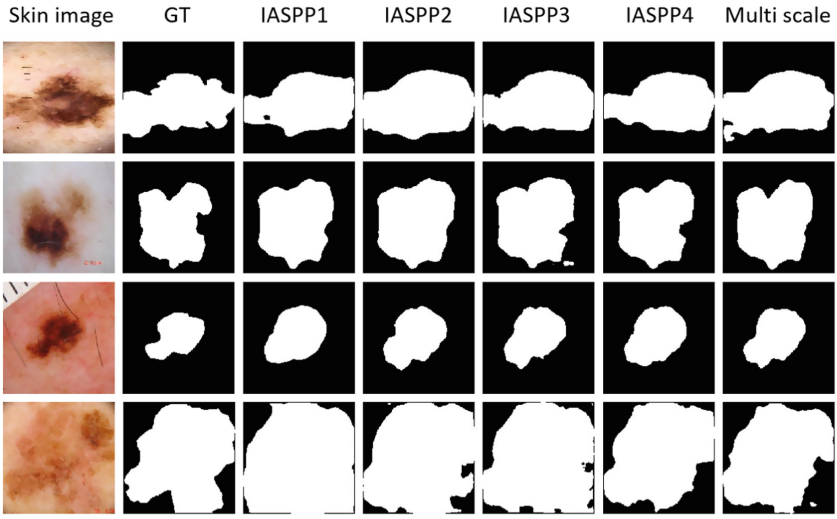



**Fig. 4.** The segmentation results are obtained by using different structures. Column 1: four dermoscopic images. Column 2: ground truth. Columns 3–7: the segmentation results of the first layer to the fourth layer and all layers embedded respectively.

- It can be observed from the third column that when extracting low-level context information, IASTrans-Net pays more attention to the significant skin lesions. For the skin lesion images with low color contrast, there is a lack of guidance to contact the deep context information (such as the fourth line segmentation results), resulting in inaccurate lesion boundary segmentation.
- Different from the previous point, the sixth column shows that context information used in the fourth layer contains more abstract semantic information. The network has a larger receptive field and can find more detailed information. Some lesion images with unclear boundaries have also been successfully segmented.
- Different from embedding the context into each layer, in the last column of experiments, our IASTrans-Net can achieve good segmentation results in significant and blurred images. Shallow context features contain local detail features, while deep context features contain rich global context information. Therefore, the combination of multi-scale context information can enhance the recognition ability of IASTrans-Net.

**Parameter Analysis**. In order to study the effect of using different dilation rates on network segmentation, a comparative experiment is carried out for analyzing the parameter sensitivity of dilation rate. The ablation rate of each layer is evaluated by DI and JA. As can be seen from Table 2, when the dilation rates of each layer is [1, 24, 36, 48], [1, 12, 24, 36], [1, 8, 12, 16], [1, 4, 6, 8], the segmentation result of the dataset is the best, the JA index is 83.79% and the DI coefficient is 90.18% . The segmentation results by using different dilation rates

**Table 2.** Quantitative experiments evaluated on ISIC2017 and ISIC2018 datasets by using different dilation rates for feature maps of different scales.

| | Feature | Rate1 | Rate2 | Rate3 | Rate4 | Rate5 |
|---|---|---|---|---|---|---|
| IASPP1 | 64 | [1, 8, 12, 16] | [1, 12, 24, 36] | [1, 24, 36, 48] | [1, 36, 48, 96] | [1, 48, 96, 12] |
| IASPP2 | 128 | [1, 6, 8, 12] | [1, 8, 12, 16] | [1, 12, 24, 36] | [1, 24, 36, 48] | [1, 36, 48, 96] |
| IASPP3 | 256 | [1, 3, 5, 7] | [1, 4, 6, 8] | [1, 8, 12, 16] | [1, 12, 24, 36] | [1,24,36,48] |
| IASPP4 | 512 | [1, 2, 3, 4] | [1, 2, 3, 4] | [1, 4, 6, 8] | [1, 6, 8, 12] | [1, 8, 12, 24] |
| 2017DI | – | 0.87576 | 0.87767 | **0.87965** | 0.87388 | 0.87693 |
| 2017JA | – | 0.80391 | 0.80099 | **0.80730** | 0.79884 | 0.80174 |
| 2018DI | – | 0.89624 | 0.89976 | **0.90186** | 0.89813 | 0.89991 |
| 2018JA | – | 0.83038 | 0.83504 | **0.83793** | 0.83345 | 0.83568 |

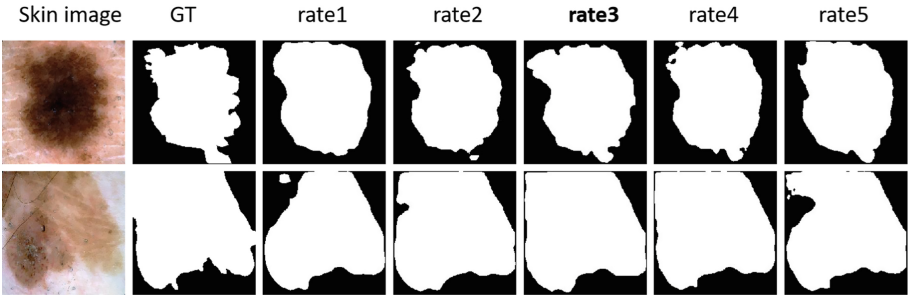



**Fig. 5.** The segmentation results obtained by using different dilation rates for the feature maps. The second column is ground truth, and the third to seventh columns are the segmentation results by using different dilation rates. Where rate3 is the dilation rate used in this paper.

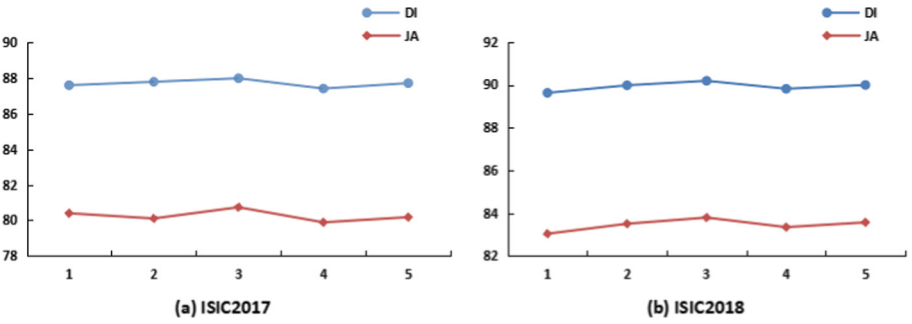



**Fig. 6.** The segmentation results with different dilation rates on ISIC2017 and ISIC2018 datasets. It can be seen from the broken line diagram in the figure, the dilation rate used in the third group can produce more accurate segmentation results than others.

for the feature maps are shown in Fig. 5. The curves of JA and DI on ISIC2017 and ISIC2018 datasets using different dilation rates are presented in Fig. 6.

### 4.4 Conclusion

In this paper, we propose a deep network IASTrans-Net to handle skin lesion segmentation. Considering the significance of multi-scale context information, we employ the atrous convolution in the designed IASPP module to capture context information and aggregate them from different scales. In addition, the parallel attention module is utilized for screening the effective information, which makes the features of different receptive fields obtained at different levels complement the feature information of local cross-channel interaction. Quantitative and qualitative experimental evaluations on both ISIC2017 and ISIC2018 datasets show the effectiveness of our proposed network.

**Acknowledgements.** This work was supported in part by the Special Project of Central Government Guiding Local Science and Technology Development (Grant No. 2021JH6/10500140), Program for the Liaoning Distinguished Professor, Program for Innovative Research Team in University of Liaoning Province (Grant No. LT2020015), the Support Plan for Key Field Innovation Team of Dalian (Grant No. 2021RT06), the Science and Technology Innovation Fund of Dalian (Grant No. 2020JJ25CY001), the Support Plan for Leading Innovation Team of Dalian University (Grant No. XLJ202010), Program for the Liaoning Province Doctoral Research Starting Fund (Grant No. 2022-BS-336).

## References

1. Al-Masni, M.A., Al-Antari, M.A., Choi, M.T., Han, S.M., Kim, T.S.: Skin lesion segmentation in dermoscopy images via deep full resolution convolutional networks. Comput. Meth. Prog. Biomed. **162**, 221–231 (2018)
2. Asadi-Aghbolaghi, M., Azad, R., Fathy, M., Escalera, S.: Multi-level context gating of embedded collective knowledge for medical image segmentation. CoRR (2020)
3. Cao, H., et al.: Swin-unet: Unet-like pure transformer for medical image segmentation. CoRR (2021)
4. Celebi, M.E., et al.: A methodological approach to the classification of dermoscopy images. Computer. Med. Imaging Graph. **31**(6), 362–373 (2007)
5. Chen, J., et al.: Transunet: Transformers make strong encoders for medical image segmentation. CoRR (2021)
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS. IEEE Trans. Pattern Anal. Mach. Intell. (2014)
7. Codella, N.C., et al.: Skin lesion analysis toward melanoma detection: a challenge at the 2017 international symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC). In: 2018 IEEE 15th International Symposium on Biomedical Imaging, pp. 168–172 (2018)
8. Day, G.R., Barbour, R.H.: Automated melanoma diagnosis: where are we at? Skin Res. Technol. **6**(1), 1–5 (2000)
9. Garnavi, R., Aldeen, M., Celebi, M.E., Bhuiyan, A., Dolianitis, C., Varigos, G.: Automatic segmentation of dermoscopy images using histogram thresholding on optimal color channels. Int. J. Med. Med. Sci. **1**(2), 126–134 (2010)

10. Hardie, R.C., Ali, R., De Silva, M.S., Kebede, T.M.: Skin lesion segmentation and classification for ISIC 2018 using traditional classifiers with hand-crafted features. CoRR (2018)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
12. Jha, D., Riegler, M.A., Johansen, D., Halvorsen, P., Johansen, H.D.: Doubleu-net: a deep convolutional neural network for medical image segmentation. In: 2020 IEEE 33rd International Symposium on Computer-based Medical Systems (CBMS), pp. 558–564 (2020)
13. Kolesnikov, A., et al.: An image is worth 16 × 16 words: Transformers for image recognition at scale (2021)
14. Liu, Z., Zerubia, J.: Skin image illumination modeling and chromophore identification for melanoma diagnosis. Phys. Med. Biol. **60**(9), 3415 (2015)
15. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
16. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical image Computing and Computer-Assisted Intervention, pp. 234–241 (2015)
17. Silveria, M., et al.: Comparison of segmentation methods for melanoma diagnosis in dermoscopy images. IEEE J. Select. Top. Sign. Process. **3**(1), 35–45 (2009)
18. Singh, V.K., et al.: FCA-Net: adversarial learning for skin lesion segmentation based on multi-scale features and factorized channel attention. IEEE Access **7**, 130552–130565 (2019)
19. Singh, V.K., et al.: FCA-Net: adversarial learning for skin lesion segmentation based on multi-scale features and factorized channel attention. IEEE Access **7**, 130552–130565 (2019)
20. Vesal, S., Ravikumar, N., Maier, A.K.: SkinNet: a deep learning framework for skin lesion segmentation. CoRR (2018)
21. Wang, H., Cao, P., Wang, J., Zaïane, O.R.: Uctransnet: rethinking the skip connections in u-net from a channel-wise perspective with transformer. CoRR (2021)
22. Wang, Z., Zou, N., Shen, D., Ji, S.: Non-local u-nets for biomedical image segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 6315–6322 (2020)
23. Wong, A., Scharcanski, J., Fieguth, P.: Automatic skin lesion segmentation via iterative stochastic region merging. IEEE Trans. Inf. Technol. Biomed. **15**(6), 929–936 (2011)
24. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881–2890 (2017)
25. Zheng, S., et al.: Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1529–1537 (2015)

# Graph Convolutional Networks (GCN)-Based Lightweight Detection Model for Dangerous Driving Behavior

Xing Wei[1,2,3(✉)], Shang Yao[1], Chong Zhao[2,4], Di Hu[2], Hui Luo[2], and Yang Lu[1,3]

[1] School of Computer and Information, Hefei University of Technology, Hefei, China
[2] Intelligent Manufacturing Institute of Hefei University of Technology, Hefei, China
[3] Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Beijing, China
[4] Engineering Quality Education Center of Undergraduate School, Hefei University of Technology, Hefei, China

**Abstract.** Real-time detection and identification of dangerous driving behaviors is an effective measure to reduce traffic accidents. Due to the high network delay, limited communication bandwidth, and weak computing power, lightweight detection models that can run on edge devices have been widely investigated and attracted considerable attention. In recent years, the Graph Convolutional Network (GCN), which models the human skeleton as a spatiotemporal graph, has achieved remarkable performance, due to its powerful capability of modeling non-Euclidean structure data. However, there are disadvantages such as the single way of extracting information, high model complexity, and inability to integrate environmental information. Therefore, we propose a lightweight dangerous driving behavior detection model based on GCN. First, two local information extraction modules are designed to extract skeleton information features. Meanwhile, we propose a multi-information fusion behavior recognition model of "people + objects" by capturing the motion information of related object. Finally, the method based on Singular Value Decomposition (SVD) rank reduction is used to compress the model to improve the speed of recognizing an action sample under sufficient detection accuracy. The proposed model respectively achieves 96% and 86.3% accuracy on the x-view benchmark of NTU-RGBD dataset and the homemade Locomotive Driver Dataset, which attains the state-of-the-art performance.

**Keywords:** Graph convolutional neural networks · Dangerous driving behavior detection · Skeletal features · Neural network compression

## 1 Introduction

In recent years, traffic safety issues have caught widespread attention from all sectors of society. To avoid traffic safety accidents caused by drivers answering the phone, smoking, or even playing with mobile phones during driving, it


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 27–39, 2022.
https://doi.org/10.1007/978-3-031-19208-1_3

is necessary to conduct real-time behavior detection of drivers. However, due to high network delay, low communication bandwidth, and other reasons, the processing results of the background server cannot be transmitted to the cab in real-time. There is an increasing need to directly analyze dangerous driving behaviors through edge computing devices in the cab environment. Still, the processing power of edge computing devices is limited, so a lightweight driving behavior analysis model is required.

Traditional human behavior detection models based on RGB video are sensitive to environmental information such as illumination and occlusion, while the dynamic human skeleton data has compact information, less redundancy, and strong robustness. However, since the skeleton data is located in the non-Euclidean geometric space, manually constructing the skeleton data directly as a sequence of joint coordinate vectors or a pseudo-image which is fed into RNNs [19] or CNNs [7] to generate the prediction, will lose the spatial information in the data. And graph convolutional network can capture the dependencies between nodes to reason from unstructured data. Currently, methods based on GCN [17] for feature extraction of human skeleton data have achieved encouraging achievements. However, the following problems remain: 1) The embedded expression process of continuously learning nodes alternately in the spatial dimension and the time dimension loses essential local information, which affects the learning and expression ability of the model. 2) The category of certain similar actions cannot be accurately judged only by the change of the joint point coordinates. For example, smoking and drinking water are highly consistent in the changes in joint coordinates. 3) GCN-based models generally require a large amount of computation. To recognize an action sample, ST-GCN [17] needs to perform 16.2 GFLOPs (Floating Point Operations), and some models even reach more than 100 GFLOPs [11].

In this paper, we study how to identify dangerous driving behaviors by integrating skeleton data with environmental semantic information, aiming to improve the speed of recognizing actions with sufficient detection accuracy. The main contributions of this work are:

1. A Multimodal Feature Graph Convolutional Neural Networks (MF-GCN) extracting local features is proposed to capture the critical spatial and temporal information in skeleton data.
2. The skeleton dataset integrates environmental semantic information to classify dangerous behaviors better.
3. We compress the model using an SVD-based approach with good results.

## 2 Related Work

### 2.1 Skeleton-Based Action Recognition

Skeleton data has received increasing concern due to its robustness to human scale, perspective, and background changes. In the work of ST-GCN proposed by Yan et al. [17], the graph convolutional neural network was applied to the

skeleton dataset for the first time. Li et al. [8] proposed an A-link inference module that extends skeleton graphs to represent higher-order joint dependencies. However, the connections between the nodes of these ST-GCN-based methods represent the natural connections of human bones, which only represent the physical structure of the human body. It leads to the representation ability being limited. Shi et al. [12] proposed a novel adaptive graph convolutional network and constructed a two-stream network 2s-AGCN using bone length and direction vector. Ye et al. [18] proposed Dynamic GCN, a new global dependency method, through which the model achieves excellent accuracy in skeleton-based action recognition. To further capture higher-order adjacency relations, Peng et al. [9] introduced higher-order Chebyshev polynomials to obtain larger receptive fields.

### 2.2 SVD Compress

The methods [4,21] of simplifying the network model by decomposing the weight matrix of the deep neural network have been widely used. Early decomposition methods [4] decompose a 4-dimensional convolution kernel into four consecutive low-rank convolutions. Still, this technique dramatically increases the number of layers in the network. Later works treat a network layer as a linear layer and then decompose this linear layer into two sub-linear layers with low rank so that the 4-D matrix tensor is reshaped into two 2-D matrices. Finally, the 2-D matrices are remapped back to convolution, resulting in two consecutive layers [3]. Zhang et al. [21] proposed channel decomposition, which uses SVD to decompose a convolutional layer with kernel size $w \times h$ into two consecutive layers with kernel size $w \times h$ and $1 \times 1$. These methods above require performing SVD at each optimization step to compute and optimize the kernel norm for each layer, but this algorithm is computationally expensive [2]. Tai et al. [16] proposed low-rank decomposition to avoid expensive SVD decomposition at each step by training the network directly with low rank from scratch, but it is necessary to manually set the size of each layer rank, which may result in suboptimal compression.

## 3 Methodology

We perform gesture estimation on the driver video and construct a spatiotemporal map on the skeleton sequence, and the extracted skeleton information is classified by our proposed MF-GCN. The classification results are fused with scene information to re-judge the driver's behavior, as shown in Fig. 1. By combining the object detection results with driver behavior classification, We can judge the dangerous action of drivers more accurately.

### 3.1 Graph Construction

We utilize PP-TinyPose to estimate the position information of 18 joints on each frame of video. PP-TinyPose is a human gesture recognition algorithm that can

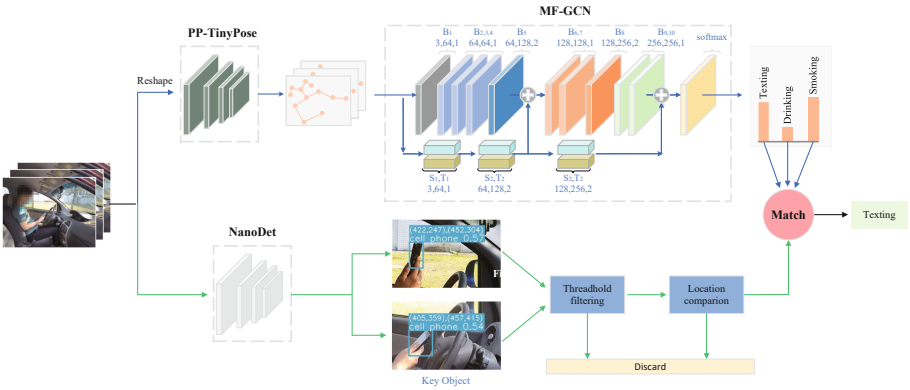



**Fig. 1.** The MF-GCN extracts the skeleton sequence's global information through ten spatiotemporal convolution blocks ($B1$-$B10$). Below the spatiotemporal convolution blocks are the spatial and temporal modules ($S1$-$S3$, $T1$-$T3$) to extract local information. The three numbers below each block represent input channels, output channels, and stride, respectively. The green line part is the scene information fusion module, which matches the classification results of MF-GCN by identifying the moving key object. (Color figure online)

be deployed on distributed devices and edge devices, which has the advantages of high precision and high speed. In the actual experiment, since the driver's legs are blocked, the main feature information of dangerous driving behaviors such as smoking and making phone calls are mostly displayed in the upper body, so the joint point information of the lower part are discarded. Only 10 important joint points of the upper body are selected to model the graph, as shown in Table 1.

**Table 1.** Ten joint points we selected.

| Number | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Joint | Right eye | Left eye | Right ear | Left ear | Right shoulder |
| Number | 6 | 7 | 8 | 9 | 10 |
| Joint | Left shoulder | Right elbow | Left elbow | Right wrist | Left wrist |

### 3.2 Local Information Capture Module Based on GCN

This paper proposes a local spatial and temporal information extraction module. Extracted local information is summed with the global information extracted in the higher-level layers. In this way, the final information obtained by the model contains not only global information, but also the rich spatiotemporal local information of the skeleton graph. The specific method training steps are given by Eq. 1:

$$f_{out} = \mathcal{W}_t(\sum_{k}^{\mathcal{B}_v} \mathcal{W}_k(f_{in} + \theta_g^l f_g^l + \theta_t^l f_t^l) \times (\mathcal{A}_k + \mathcal{G}_k)) \tag{1}$$

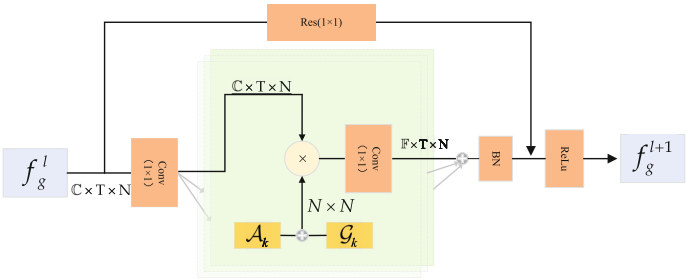



**Fig. 2.** Schematic diagram of the spatial module structure.

where $\mathcal{B}_v$ denotes the size of the spatial convolution kernel, $k$ means the specific spatial convolution kernel, $\mathcal{W}_k \in \mathcal{R}^{\mathbb{C}\times\mathbb{C}\times1\times1}$ and $\mathcal{W}_t \in \mathcal{R}^{\mathbb{F}\times\mathbb{C}\times9\times1}$ ($\mathbb{C}$ means output channel, $\mathbb{F}$ means input channel) represent the spatial convolution weight matrix and the temporal convolution weight matrix, respectively, $\mathcal{A}_k \in \mathcal{R}^{N\times N}$ represents the adjacency matrix, $N$ is the number of joint points, $\mathcal{G}_k$ is the adaptive adjacency matrix with the same dimensions as $\mathcal{A}_k$, and will be parameterized and optimized along with other parameters during training; $\theta_t^l, \theta_g^l \in \{0, 1\}$ control input $f_t^l, f_g^l$ of the time module and the spatial module of the $l$-th respectively. $f_{in}$ represents the feature map of the skeleton map extracted by a spatiotemporal convolutional block. Each node aggregates the information of spatial neighbor nodes and the information of temporal neighbor nodes.

Compared with the spatiotemporal convolution block, the spatial module $S$ aims to extract only local spatial information. A feature map is decomposed into three sub-map through the convolution kernel of $\mathcal{K}_1 \in \mathcal{R}^{3\mathbb{C}\times\mathbb{C}\times1\times1}$, and each submap expresses action features of different scales, as shown in Fig. 2. To match the output channels of the spatiotemporal convolution block, the convolution kernel $\mathcal{K}_2 \in \mathcal{R}^{\mathbb{F}\times\mathbb{C}\times1\times1}$ is used to change the dimension of the feature map. Finally, the convolution of the spatial dimension on the skeleton graph is realized by aggregating the information of the neighbor nodes, and the aggregation method is given by Eq. 2:

$$f_g^{l+1} = \sum_k^{\mathcal{B}_v} \left( \left( \mathcal{W}_k^1 f_g^l \right) \times \left( \mathcal{A}_k + \mathcal{G}_k \right) \right) \times \mathcal{W}_k^2 \tag{2}$$

where $f_g^l$ represents the feature map after the $l$-th aggregation of the spatial module and $\mathcal{W}_k^1, \mathcal{W}_k^2$ are the weight matrices of the convolution kernels $\mathcal{K}_1, \mathcal{K}_2$ respectively.

The time module $T$ is similar to the spatial module and aims to extract local time information. $T$ aggregate the information of neighbor nodes in the time dimension through the convolution kernel $\mathcal{K} \in \mathcal{R}^{\mathbb{F}\times\mathbb{C}\times3\times1}$. The specific details follow Eq. 3:

$$f_t^{l+1} = W_t f_t^l \tag{3}$$

where $f_t^l$ represents the feature map after the $l$-th aggregation of the time module and $\mathcal{W}_t$ is the weight matrix of the convolution kernel $\mathcal{K}$.

### 3.3 Scene Fusion Based on Lightweight Object Detection

In this section, for the situation where different behaviors may have highly consistent changes in joints, through the fusion of the multi-information of people and objects, the driver's behavior can be judged more accurately. Usually, the object's spatial position associated with the action will change as the action progresses, such as making a phone call, drinking water, etc. We use NanoDet to detect key targets in driving scenarios. NanoDet is a lightweight target detection model that has the advantages of fast speed, a small amount of calculation, and model parameters. We extract 21,000 images from the Locomotive Driver Dataset by drawing frames which divided into four categories, as shown in Table 2. We use these images to train NanoDet.

**Table 2.** Key objects associated with the action.

| Number | C01 | C02 | C03 | C04 |
|---|---|---|---|---|
| Joint | Cigarette | Water glass | Cell phone | Bottle |

The cigarette corresponds to the smoking action in the Locomotive Driver Dataset, the water glass and bottle correspond to the drinking action, and the cell phone corresponds to the action category of talking on the phone or texting. In the actual detection, the trained NanoDet network is used to detect the video frame by frame, and the key object is marked with a label and 2D coordinates. The specific process is shown in the green line part in Fig. 1. Notably, a thing will be discarded if it is stationary.The detected target object is matched with the classification results of MF-GCN to re-judge the driver's behavior.

### 3.4 SVD-Based Compression

For the MF-GCN network that we have trained, we use the SVD-based method to compress the model.

For a convolutional layer with input channel $\mathbb{F}$, output channel $\mathbb{C}$, and kernel size $k_1 \times k_2$, we can interpret the layer $\mathcal{K} \in \mathcal{R}^{\mathbb{F}\times\mathbb{C}\times k_1\times k_2}$ as a linear layer $\hat{\mathcal{K}} \in \mathcal{R}^{\mathbb{F}\times\mathbb{C}k_1k_2}$ and the corresponding rank $j$-approximation as two subsequent linear layers of shape $\mathbb{F}\times j$ and $j\times\mathbb{C}k_1k_2$. Mapped back to convolutions, this corresponds to a $\mathcal{K}_2 \in \mathcal{R}^{j\times\mathbb{C}\times k_1\times k_2}$ convolution followed by a $\mathcal{K}_1 \in \mathcal{R}^{\mathbb{F}\times j\times 1\times 1}$ convolution. But performing the decomposition of SVD is computationally expensive. In order to avoid repeating the SVD operation at each step, we perform SVD full-rank decomposition on $\hat{\mathcal{K}}$ to get $\mathcal{M} \in \mathcal{R}^{\mathbb{F}\times j}$, $\mathcal{N} \in \mathcal{R}^{\mathbb{C}k_1k_2\times j}$, $s \in \mathcal{R}^j$. The weight matrices of $\mathcal{K}_1$ and $\mathcal{K}_2$ are reconstructed from $\mathcal{M}diag\left(\sqrt{s}\right)$ and $diag\left(\sqrt{s}\right)\mathcal{N}^T$, directly. In the SVD training process, each layer uses the decomposed variable $\mathcal{M}, s, \mathcal{N}$ instead of the original convolution kernel or weight matrix. The forward pass is by converting $\mathcal{M}, s, \mathcal{N}$ into two consecutive network layers, and

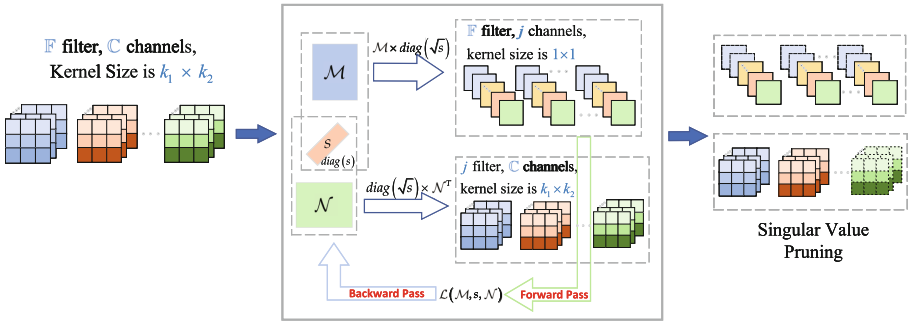



**Fig. 3.** Schematic diagram of the SVD compression structure. A $\mathcal{K} \in \mathcal{R}^{\mathbb{F}\times\mathbb{C}\times k_1\times k_2}$ convolution layer is decomposed into the product of two convolution kernels $\mathcal{K}_1 \in \mathcal{R}^{\mathbb{F}\times j\times 1\times 1}$ and $\mathcal{K}_2 \in \mathcal{R}^{j\times\mathbb{C}\times k_1\times k_2}$. Some redundant convolution kernels are filtered by the rank reduction method. The part of the dotted line on the right represents the cropped convolution kernel.

both backward pass and optimization work directly on $\mathcal{M}, s, \mathcal{N}$. As shown in Fig. 3. In this way, we can obtain $s$ without performing expensive SVD operations at each step of training. When the singular vector matrix $\mathcal{M}, \mathcal{N}$ is already orthogonal, reducing the rank of the decomposition network is equivalent to making the singular value vector $s$ of each network layer as sparse as possible. We choose an invariant modulus Eq. 4 proposed in the work of Dilip Krishnan et al. [6] to represent sparsity:

$$\mathcal{L}^h(s) = \frac{\|s\|_1}{\|s\|_2} = \frac{\sum_i |s_i|}{\sqrt{\sum_i s_i^2}} \tag{4}$$

The regularization operator is differentiable almost everywhere and has scale invariance. Based on the above analysis, we propose the overall objective function Eq. 5 for decomposing training:

$$\mathcal{L}(\mathcal{M}, s, \mathcal{N}) = \mathcal{L}_T + \lambda_h \sum_{l=1}^{B} \mathcal{L}^h(s) \tag{5}$$

where $\mathcal{L}_T$ is the training loss on the decomposed network layers, $B$ is the total number of network layers and $\lambda_h$ is the decay parameter that can be traded off between accuracy and FLOPs to get a low-rank model. Finally, a lightweight graph convolutional network is obtained.

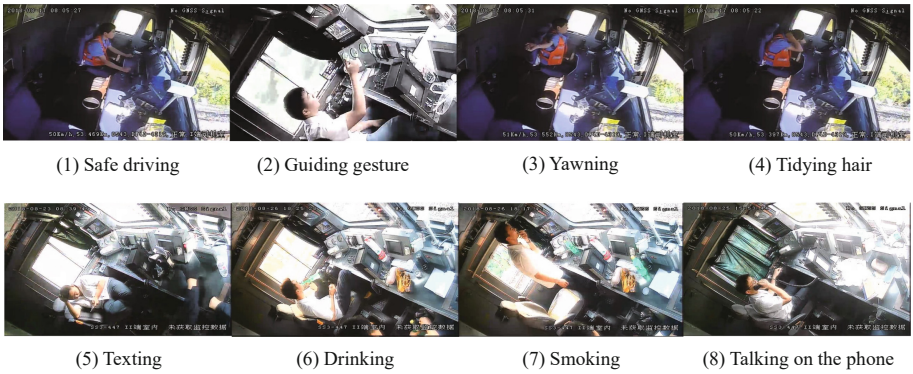



**Fig. 4.** Examples of 8 categories in the Locomotive Driver Dataset. There are two types of normal driving and six categories of dangerous driving. The original video resolution is $1280 \times 720$, 13 fps.

## 4 Experiments

### 4.1 Dataset and Parameter Settings

**NTU-RGBD**: NTU-RGBD [10] is currently the largest and most widely used multimodal indoor behavior recognition dataset. The original paper of the dataset recommends two benchmarks: 1) Cross-subject (CS) and 2) Cross-view (CV).

**Locomotive Driver Dataset** : We photographed specific datasets for train driver driving situations, which are divided into eight categories. As shown in Fig. 4. There are two normal driving behaviors and six dangerous driving behaviors. Our dataset was filmed with eight drivers. To ensure the diversity of our data, we selected participants with different heights, weights, and different driving styles, wearing different uniforms. We process each action clip into a video of about 3 s, about 13 frames per second. Our dataset contains a total of 9362 instances, of which we use 6553 instances for training and 2809 instances for testing.

**Parameter Settings** : All our experiments are implemented on the PyTorch deep learning framework, using stochastic gradient descent (SGD) with Nesterov momentum (0.9) as the optimization strategy, cross-entropy as the loss function, weight decay set to 0.0004, and batch size set to 90.

**Table 3.** Comparison with 11 state-of-the-art methods on NTU-RGBD dataset in terms of accuracy, parameters ($\times 10^6$) and FLOPs ($\times 10^9$). MF-GCN (compressed) represents the result after compression using SVD.

| Methods | Acc (CS) | Acc (CV) | Param | FLOPs |
|---|---|---|---|---|
| ST-GCN [17] | 81.5 | 88.3 | 3.1 | 16.32 |
| AS-GCN [8] | 86.8 | 94.2 | 9.5 | 26.76 |
| RA-GCNv2 [15] | 87.3 | 93.6 | 6.21 | 32.8 |
| 2s-AGCN [12] | 88.5 | 95.1 | 6.94 | 37.32 |
| NAS-GCN [9] | 89.4 | 95.7 | 6.57 | – |
| 2s-AGC-LSTM [13] | 89.2 | 95 | 22.89 | 54.4 |
| SR-TSL [14] | 84.8 | 92.4 | 19.07 | 4.2 |
| Clips+CNN+MTLN [1] | 79.57 | 84.83 | 62 | – |
| C-MANs [7] | 83.72 | 93.8 | 28.4 | – |
| TS-SAN [5] | 87 | 92.4 | – | – |
| VA-LSTM [19] | 79.4 | 87.6 | – | – |
| MF-GCN | 89.8 | 96 | 2.56 | 12.76 |
| MF-GCN (compressed) | 89.5 | 95.6 | 1.86 | 6.82 |

### 4.2 Comparison with the State-of-the-Art

On the Locomotive Driver Dataset and NTU-RGBD dataset, we compare the final model with skeleton-based action recognition methods, which including CNN-based methods [1,7], LSTM-based methods [13,14,19], self-attention-based methods [5], and GCN-based methods [8,9,12,15,17,20]. The results of the comparison are shown in Tables 3 and 4. Since the scene information fusion module is especially proposed for the Locomotive Driver Dataset, the model performance of fused and unfused scene information are compared on this dataset. As shown in Table 4, the results show that the model combined with the scene information is obviously superior to GCN-based models alone. On the NTU-RGBD dataset, we compared the model's accuracy and complexity with the state-of-the-art algorithms, and the results are presented in Table 3. Part of the data are calculated from the code they posted. To balance the loss of accuracy and model complexity, we employ SVD with a compression ratio of 2×. On both data, our model shows advanced performance. Furthermore, the SVD-based method effectively compresses the number of parameters and calculation of the model.

**Table 4.** Comparison with other methods on the Locomotive Driver Dataset, "MF-GCN+obj" represents MF-GCN fused with environmental semantic information

| Methods | Top-1 Accuracy |
|---|---|
| ST-GCN [17] | 80.2 |
| VA-LSTM [19] | 75.6 |
| 2s-AGCN [12] | 83.5 |
| SGN [20] | 84.4 |
| NAS-GCN [9] | 84.7 |
| MF-GCN | 85.4 |
| MF-GCN+obj | 86.3 |

**Table 5.** Performance comparison of the effectiveness of the spatial module ($S$) and the temporal module ($T$) on the Locomotive Driver Dataset, de/$X$ means removing the $X$ module.

| Methods | Top-1 Accuracy |
|---|---|
| 2s-AGCN [12] | 83.5 |
| 2s-AGCN $+S$ | 84.2 |
| 2s-AGCN $+T$ | 84.5 |
| 2s-AGCN $+S+T$ | 85.0 |
| MF-GCN de/$T$ | 84.5 |
| MF-GCN de/$S$ | 84.9 |
| MF-GCN | 85.4 |

## 5 Ablation Study

We verify the effectiveness of our proposed algorithm by conducting ablation experiments on the Locomotive Driver Dataset and NTU-RGBD dataset, and adopt 2s-AGCN [12] as our experimental benchmark.

### 5.1 Significance of Local Information

According to the introduction in Sect. 3.2, there are two main sources of local information: temporal module ($T$) and spatial module ($S$). To explore the effectiveness of the two modules, we add two modules to 2s-AGCN and remove two modules from our model, respectively. The result shows in Table 5. The results show that adding two modules to 2s-AGCN [12] improves the accuracy by 0.84% ($S$) and 1.20% ($T$), respectively, and removing two modules from MF-GCN loses the accuracy by 1.05% ($T$) and 0.59% ($S$), respectively. It is proved that the extraction of local information does help to improve the accuracy of action classification.

### 5.2 Effectiveness of Model Compression

In NTU-RGBD dataset x-view benchmark and Locomotive Driver Dataset. We compare the impact of different compression rates on the accuracy results. The results are shown in Fig. 5. It can be seen from the experimental results that the SVD-based compression algorithm can effectively compress the model complexity while having less damage to the accuracy.

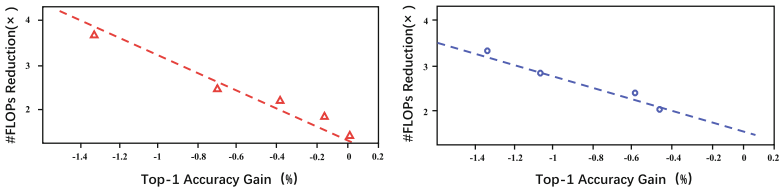



**Fig. 5.** The left shows the relationship between the FLOPs compression ratio and accuracy loss on the NTU-RGBD dataset by using SVD compression in MF-GCN model, and the right shows the relationship on the Locomotive Driver Dataset.

### 5.3 Visualization

We select some actions that are difficult to be recognized well and draw their confusion matrices on 2s-AGCN and MF-GCN. As shown in Fig. 6, the vertical axis represents the category of our actual action, and the horizontal axis represents the predicted action category label. Since there are many classification categories in NTU-RGBD, we selected some of the action categories (10–31). Fig. 6, there are two groups of similar actions; the first group is reading, writing, playing tablet, and typing, which are framed by the red matrix. From the perspective of the skeleton, these actions are all done by the slight shaking of the joints of the hands and elbows, and the changes in the joints of these actions are very similar in space and time. The second group is surrounded by a yellow matrix, which contains two sets of actions put on glasses and take off glasses; this group of activities has similar temporal dynamics but differs in spatial configuration. It can be seen that the recognition accuracy of this category on both 2s-AGCN and our MF-GCN is greater than the actions of the previous group. However, from the perspective of the confusion matrix, we can also see that the recognition accuracy of these similar actions is still not high enough only from the changes of joint points position. Overall, our model has achieved a significant improvement compared to 2s-AGCN.

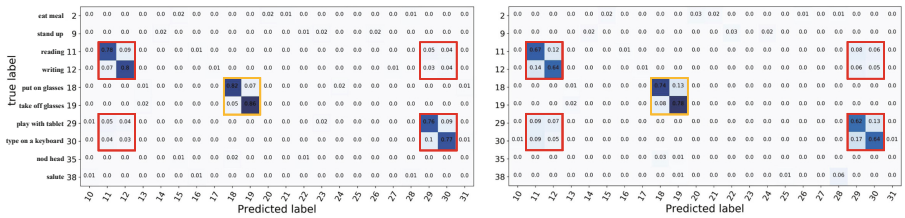



**Fig. 6.** Confusion matrices of MF-GCN (left) and 2s-AGCN (right), where the numbers on the axes represent the index of each action category, red and yellow rectangles denote two groups of similar actions.

## 6 Conclusion

This paper proposes a new model MF-GCN for skeleton-based action recognition, which extracts different feature information from three different paths so that the model finally learns that the feature map contains rich global and local information. Then the SVD-based method is used to simplify the model, and a lightweight and high-precision GCN model is obtained. In addition, aiming at the problem that the GCN model is prone to errors in identifying some similar actions. We define the "human + object" multi-information fusion method to re-identify human behavior, thereby further improving the performance. The final model is evaluated on the large-scale action recognition dataset NTU-RGBD and the homemade Locomotive Driver Dataset, achieving state-of-the-art performance on both datasets.

**Acknowledgements.** This work was supported by Joint Fund of Natural Science Foundation of Anhui Province in 2020 (2008085UD08), Anhui Provincial Key R&D Program (202004a05020004), Open fund of Intelligent Interconnected Systems Laboratory of Anhui Province (PA2021AKSK0107), Intelligent Networking and New Energy Vehicle Special Project of Intelligent Manufacturing Institute of HFUT (IMIWL2019003, IMIDC2019002).

## References

1. Cho, S., Maqbool, M., Liu, F., Foroosh, H.: Self-attention network for skeleton-based human action recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 635–644 (2020)
2. Giles, M.: An extended collection of matrix derivative results for forward and reverse mode automatic differentiation (2008)
3. Idelbayev, Y., Carreira-Perpinán, M.A.: Low-rank compression of neural nets: learning the rank of each layer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8049–8059 (2020)
4. Jaderberg, M., Vedaldi, A., Zisserman, A.: Speeding up convolutional neural networks with low rank expansions. arXiv preprint arXiv:1405.3866 (2014)
5. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: A new representation of skeleton sequences for 3D action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3288–3297 (2017)
6. Krishnan, D., Tay, T., Fergus, R.: Blind deconvolution using a normalized sparsity measure. In: CVPR 2011, pp. 233–240. IEEE (2011)
7. Li, C., Xie, C., Zhang, B., Han, J., Zhen, X., Chen, J.: Memory attention networks for skeleton-based action recognition. IEEE Transactions on Neural Networks and Learning Systems (2021)
8. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: NTU RGB+D: a large scale dataset for 3D human activity analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1010–1019 (2016)
9. Peng, W., Hong, X., Chen, H., Zhao, G.: Learning graph convolutional network for skeleton-based human action recognition by neural searching. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 2669–2676 (2020)

10. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: NTU RGB+D: a large scale dataset for 3D human activity analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1010–1019 (2016)
11. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Skeleton-based action recognition with directed graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7912–7921 (2019)
12. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12026–12035 (2019)
13. Si, C., Chen, W., Wang, W., Wang, L., Tan, T.: An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. In: proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1227–1236 (2019)
14. Si, C., Jing, Y., Wang, W., Wang, L., Tan, T.: Skeleton-based action recognition with spatial reasoning and temporal stack learning. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 103–118 (2018)
15. Song, Y.F., Zhang, Z., Shan, C., Wang, L.: Richly activated graph convolutional network for robust skeleton-based action recognition. IEEE Trans. Circuits Syst. Video Technol. **31**(5), 1915–1925 (2020)
16. Tai, C., et al.: Convolutional neural networks with low-rank regularization. arXiv preprint arXiv:1511.06067 (2015)
17. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI Conference on Artificial Intelligence (2018)
18. Ye, F., Pu, S., Zhong, Q., Li, C., Xie, D., Tang, H.: Dynamic GCN: context-enriched topology learning for skeleton-based action recognition. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 55–63 (2020)
19. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2117–2126 (2017)
20. Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., Zheng, N.: Semantics-guided neural networks for efficient skeleton-based human action recognition. In: proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1112–1121 (2020)
21. Zhang, X., Zou, J., He, K., Sun, J.: Accelerating very deep convolutional networks for classification and detection. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 1943–1955 (2015)

# Increasing the Accuracy of Secure Model for Medical Data Sharing in the Internet of Things

Junhua Wu, Huiru Zhang, Guangshun Li(✉), and Kan Yu

School of Computer Science, Qufu Normal University, Rizhao, China
shdwjh@163.com, guangshunli@qfnu.edu.cn

**Abstract.** The security of medical data sharing (MDS) plays an important role in the area of healthcare. Significantly, achieving its security faces more challenges due to the feature of multiparty holding, higher complexity, and serious data silos. Different from traditional secure schemes, which established model cannot deal with the above three problems due to the low accuracy of the MDS secure model, this paper designs a novel secure MDS model and two schemes to increase the accuracy of the model. In detail, to eliminate the issues of data silos and point failure, we combine the federated learning (FL) with blockchain technology into MDS secure model, and the data confidentiality of the exchanged data in the process of FL can be further ensured by differential privacy (DP). Then, to increase the accuracy of the secure MDS model, we design a validation incentive mechanism based on model quality (VIM) and an effective DP method with assigned weights (AWDP), in terms of participants' enthusiasm and noise accumulation, respectively. Simulations show that the established model is effective and correct and the designed VIM and AWDP can achieve higher accuracy than current popular methods, resulting in 30% increment.

**Keywords:** Data sharing · Federated learning · Blockchain · Model quality · Incentive mechanism

## 1 Introduction

With the development of Internet of Things technology and the improvement of medical informatization, there is an increasing demand for data utilization and sharing among hospitals, patients and researchers [1]. MDS can improve the quality and safety of medical services. However, the relative independence of each department and the prevalence of low data quality, data silos, and poor data circulation and sharing make MDS more challenging [2]. Therefore, there is an urgent need for efficient mechanisms or methods to balance data sharing and privacy protection, break down information silos, and improve the security of MDS.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 40–51, 2022.
https://doi.org/10.1007/978-3-031-19208-1_4

To achieve secure data sharing, many studies have utilized the features of blockchain decentralization, joint maintenance by multiple parties, and traceability, combined with cryptographic techniques to ensure traceable, non-tamperable, and non-repudiation of transactions [3]. However, the incompleteness and high sensitivity of medical big data impose more stringent requirements on the security and accuracy of shared data. For secure MDS, many researchers have always stored data locally by introducing FL [4,5]. However, these approaches ignore the accuracy and availability of shared medical data.

In summary, the challenges to addressing MDS security are: 1) Third-party servers may analyze participants' original information by collecting their data; 2) Large DP noise can seriously affect model accuracy and performance [6]; 3) It is challenging to obtain highly accurate data samples while ensuring that patients' private information is not compromised.

To address the above challenges, this paper investigates how to maximize the usability of the model while achieving the security of medical data sharing. The main contributions are summarized as follows:

1. A secure MDS model based on consortium blockchain and FL is established. Participants break down data silos through joint modeling without sharing private data.
2. An effective DP method for assigning weights is designed. Giving different update weights to different noises can prevent the gradient information of the aggregated model from being overwhelmed by the accumulated noise.
3. We design a validation incentive mechanism based on model quality to verify whether the models provided by the participants are qualified or not. This mechanism increases the motivation of participants and ensures the high accuracy and usability of the training models.

The rest of this article is organized as follows. In Sect. 2, we briefly review the related work of this paper. Section 3 illustrates the system model we designed. Section 4 introduces the VIM and the secure federated averaging (SFA) algorithm proposed in this paper in detail. Section 5 conducts security analysis and performance evaluation on our method. Finally, we conclude this article in Sect. 6.

## 2 Related Work

In this section, we review the research related to this paper based on techniques such as FL, DP, and incentive mechanisms. In the architecture of FL, achieving data sharing does not require transferring data between them, but using datasets for model training, reducing the risk of user privacy data leakage [7,8]. Based on the advantages of FL in privacy protection, many studies have introduced solutions combining FL, DP, and blockchain to ensure data privacy and network security. The advantage of DP is that it can effectively protect private information without caring about the background knowledge possessed by the attacker [9].

In [10], to build a smart home system, Zhao *et al.* designed a hierarchical crowdsourcing FL system that uses a reputation mechanism combined with blockchain to prevent malicious model updates. In addition, it proposed a new standardization technique and used DP to prevent the leakage of sensitive customer information and improve the accuracy of testing [11]. Designed a blockchain-based FL framework for comprehensive defense against poisoning attacks and introduced partial DP to resist membership inference attacks and improve the security of FL in 5G networks. In [12], Jiang *et al.* designed a privacy-enhanced FL framework to resist spoofing attacks based on membership proofs and cryptographic accumulators in public blockchains. The study also designed a result verification algorithm based on ElGamal encryption for verifying the correctness of aggregation results, which improved the robustness and accuracy of the model. The above work considered the single point of failure problem in FL and also the possibility of malicious actors inferring the original private data of the user through the model parameters. The advantages of using DP are well demonstrated, but both ignored the impact of noise on model accuracy [13]. How to ensure the high accuracy of the model while protecting data privacy is a question worth exploring.

Incentive mechanisms are one of the cores of blockchain and FL, which can be classified into three categories based on pricing strategies: reputation-based, data quality-based and auction-based [14]. For the mobile crowdsensing incentive task, [15] proposed an incentive mechanism that balances data quality and privacy protection based on a zero-knowledge model for data reliability assessment, which ensured data reliability while protecting data privacy. In [16], data sharing for internet of vehicles applications is of great interest, and Chen *et al.* designed an auction-based incentive mechanism using a consortium blockchain to motivate users to participate in collecting and sharing data while guaranteeing the security of both on-chain and off-chain data to ensure the high quality and trustworthiness of the shared data.

However, existing incentive mechanisms are difficult to meet the demand for security in distributed environments, such as vulnerability to Sybil attacks. In Sybil attacks, a single attacker can forge multiple identities to exist in the network, thus interfering with normal network activities and reducing the security of the network. In summary, being able to defend against malicious attacks while ensuring model accuracy is the main challenge of current research.

## 3 System Model

Secure MDS model based on consortium blockchain and FL is shown in Fig. 1.

In secure MDS model, the user who wants to participate in the task obtains medical health data through sensors and other devices and sends a request for an initial model, which in turn is trained as a medical model. After the local model training, the model parameters are protected using the AWDP method. Before the model is added to the blockchain as a transaction, the model quality is verified using the VIM designed in this paper. Using the blockchain to store

and distribute the global model reduces the possibility of malicious corruption by third-party servers. After the verification is passed, it is uploaded to the blockchain and aggregated and averaged using the SFA algorithm to obtain models with high accuracy and good security performance.

The symbol description in this paper is shown in Table 1.

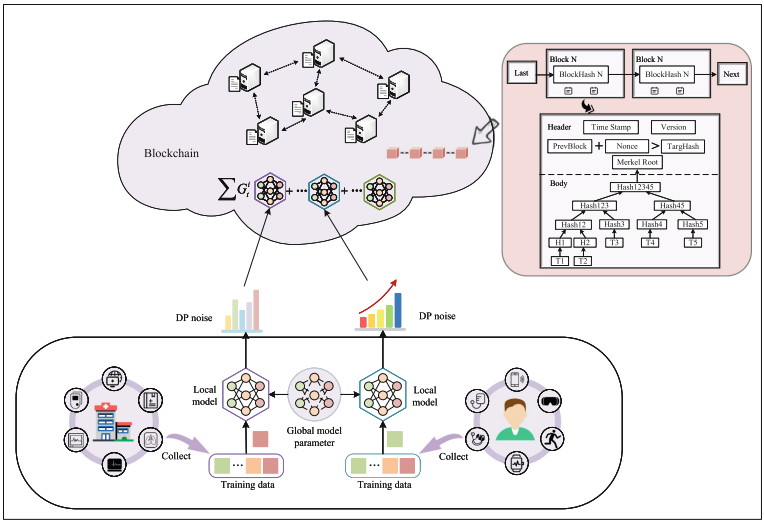



**Fig. 1.** Secure MDS model.

## 4 VIM and SFA Algorithm

### 4.1 Validation Incentive Mechanism Based on Model Quality

The members of the verification team formulate and publish a model quality judgment standard and corresponding rewards in advance. The node that wants to publish the task becomes the task publisher and broadcasts the task information, which includes: the specific content of the task, the time limit requirement, $Bgr$, the requirement for the historical quality, and the basic requirement for the task data.

**Participants Selection.** The historical quality of completing the task is the main focus of selecting participants. We adopt the idea of time window and exponential decay, and $HoT$ of completing the task is calculated by

$$HoT = baseH + \sum_{i=1}^{m} (e^{-\lambda_i} * \frac{1}{m_i} \sum_{j=1}^{m_i} (Q_{i,j} - baseH)).$$

**Table 1.** The symbol description

| Symbol | Meanings | Page |
|---|---|---|
| $Bgr$ | The budget requirement | 4 |
| $HoT/baseH$ | The historical quality/basic value of $HoT$ | 4 |
| $m$ | The number of time windows | 4 |
| $\lambda_i$ | The lower bound of the time window | 4 |
| $m_i$ | The number of tasks completed by participants | 4 |
| $Q_{i,j}$ | The complete quality of the task | 4 |
| $NoC$ | The selection number | 5 |
| $\alpha$ | The historical quality as a percentage of participant selection | 5 |
| $VsR/baseV$ | The verification success rate/basic value of $VsR$ | 6 |
| $N_i^{cor}$ | The frequency of correct verification by the verifiers | 6 |
| $N_i^{total}$ | The total number of verifications of the node | 6 |
| $\beta$ | The proportion of historical quality in validator selection | 6 |
| $\rho_i$ | The ratio of the reward | 7 |
| $dnum$ | The amount of data received by the verifiers | 7 |
| $\bar{w_t}$ | The latest model parameter | 7 |
| $K/n$ | The number of participants/training data | 7 |
| $\eta$ | The learning rate | 7 |
| $g_k^{(b)}$ | The gradient | 7 |
| $w_{t+1}^{(k)^*}$ | The latest model parameter added to DP | 7 |
| $S/\varepsilon$ | The privacy sensitivity/budget of DP | 7 |
| $q$ | The weight positively correlated with the amount of noise | 7 |
| $w'$ | The model parameter after weighting | 7 |
| $w_0$ | The initial model parameter | 8 |
| $C_t$ | A set of randomly selected participants | 8 |
| $\rho$ | The percentage of users calculated in each round | 8 |
| $P_k$ | An index set of data points at uer $k$ | 8 |
| $w_{1,1}^{(k)}$ | The latest model parameter obtained from the server | 8 |
| $I/B$ | The number of iterations/batches | 8 |
| $D_k$ | A data set owned by the $k$th participant | 8 |
| $M/b/t$ | The batch/batch number/training round | 8 |
| $w$ | The final aggregated global model parameter | 8 |

In the initial situation, all nodes that want to do the task for the first time can be selected, We set $NoC$ for the candidate node. The initial selection times of all candidate nodes are 0, and the selection times are changed accordingly according to the number of times they are selected. Therefore, the participants'

choices are as follows:

$$PaR = HoT * \alpha + NoC * (1 - \alpha).$$

**Verifiers Selection.** We select verifiers based on the historical quality of the tasks completed by the node and the success rate of verification. Using the idea of calculating $HoT$, the $VsR$ of the verifiers is calculated as

$$VsR = baseV + \sum_{i=1}^{m} (e^{-\lambda_i} * \frac{N_i^{cor}}{N_i^{total}}).$$

The verifiers' selection is shown in (1). The verifiers verify the data model by polling. Once a verifier fails to complete the verification within the specified time, its verification accuracy rate as a verifier will decrease, and the next verifier will take over the verification. The verification results are used to generate a Merkle tree and record it in the blockchain.

$$VeR = HoT * \beta + VsR * (1 - \beta). \tag{1}$$

In this process, we design the following method to select the initial verifier. We select initial verifiers from all verification candidates by randomly drawing lots to ensure the unpredictability and randomness of the initial verifier. If the block network simply uses a random lottery to conduct elections, the attacker can generate a large number of false nodes at a very low cost, which will make the system vulnerable to witch attacks.

To prevent Sybil attacks, each node monitors the behavior of other nodes by maintaining a monitoring table. Nodes only forward blocks of specific users will be quickly identified, blacklisted, and notified to other nodes. The monitoring table includes the receiving block number, the node address, and the count of the sending block. All nodes in the blockchain network have a public suspicious list. If the count on the node monitoring table exceeds a certain threshold, the node will put the corresponding address in the public suspicious list. When a node receives a block, it checks whether the node address in the block header is consistent with the address in the public suspicious list, and then decides whether to forward the block to other nodes. To prevent the normal node from being "unjustified", the node can also get rid of the public list by forwarding the blocks of the normal node fairly.

**Validation of the Model.** Firstly, according to the basic requirements of the task data, it is divided into three sets $A$, $B$, and $C$ (set $A$ represents the most satisfactory requirements, and decreases in order). These three sets reflect the degree of conformity of the results to the task. This method used the idea of fuzzy mathematics. Here, three sets are membership functions. The boundary point $\varphi$ between $C$ and $B$ and the boundary point $\xi$ between $B$ and $A$ need to be determined. We use the rule of thirds to determine the three-phase membership function.

Suppose $P = \{A, B, C\}$, each $F$ test determines a division of $U$, and each division determines a pair of numbers $(\xi, \varphi)$. $\xi$ and $\varphi$ obey a normal distribution. $(\xi, \varphi)$ determines the mapping $e(\xi, \varphi)$:

$$e(\xi,\varphi)(x) = \begin{cases} A(x) & x \le \xi, \\ B(x) & \xi < x \le \varphi, \\ C(x) & x > \varphi. \end{cases}$$

Calculate according to the probability method:

$$A(x) = 1 - \Phi\left(\frac{x-\mu_1}{\sigma_1}\right), C(x) = \Phi\left(\frac{x-\mu_2}{\sigma_2}\right),$$

$$B(x) = 1 - A(x) - C(x) = \Phi\left(\frac{x-\mu_1}{\sigma_1}\right) - \Phi\left(\frac{x-\mu_2}{\sigma_2}\right).$$

Here,

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt.$$

We use the similarity of the model data in the same area in the task, use the Euclidean distance to calculate the similarity of the two models, and then perform bottom-up hierarchical clustering. According to the result of hierarchical clustering, the cluster with the largest amount of resultant data is regarded as the position where the correct model exists. The center model $\gamma_m$ and the boundaries $\gamma_s$ and $\gamma_l$ can be obtained. Based on $\gamma_m$, $\gamma_s$, and $\gamma_l$, the model quality range is divided into three sets, $A$, $B$, and $C$. These three sets reflect the distance between the model and the correct model. Taking these two factors into consideration, the model quality level is comprehensively evaluated and divided into five levels. Finally, corresponding rewards are given to each level corresponding to the task completer as

$$reward_i = \rho_i * \frac{Bgr}{dnum}, i = 1, 2, 3, 4, or\, 5.$$

### 4.2 SFA Algorithm

Algorithm 1 gives the pseudo-code of the SFA algorithm. Next, we introduce the main phases of the SFA algorithm.

*Step* 1: Blockchain nodes broadcast $w_0$ to all participating nodes and select a set of participants.

*Step* 2: All participants conduct model training in parallel, and the participant calculate $w_{t+1}$ according to (2). Each participant uses local data to perform gradient descent on $\bar{w_t}$ according to (3) and adds $w_{t+1}^{(k)}$ to Laplace noise through (4).

$$w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^{K} \frac{n_k}{n} g_k. \tag{2}$$

$$\forall k, w_{t+1}^{k} \leftarrow \bar{w_t} - \eta g_k. \tag{3}$$

**Algorithm 1.** Secure Federated Averaging (SFA)

**Input:** $K$, $w_0$, $I$

**Blockchain Node:**
initialize $w_0$ and broadcast to all participants
**for** $t = 1, 2, ...$ **do**
  $C_t$ = (random set of $\max(K\rho, 1)$ clients)
  **for** each participant $k \in C_t$ **do** in parallel
    $w_{t+1}^{(k)^*} \leftarrow$ParticipantUpdate$(k, \bar{w_t})$
  **end for**
  Validation nodes use validation incentive mechanism to validate models
  **if** verified **then**
    $\bar{w_{t+1}} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w'$
  **else**
    return false
  **end if**
**end for**
**ParticipantUpdate**$(k, \bar{w_t})$**:**
$w_{1,1}^{(k)} = \bar{w_t}$
**for** local iteration $i$ from 1 to $I$ **do**
  batches$\leftarrow$(Randomly divide the data set $D_k$ into the size of batch $M$)
  $w_{1,i}^{(k)} = w_{B,i-1}^{(k)}$
  **for** $b$ from 1 to $B = \frac{n_k}{M}$ **do**
    Calculate $g_k^{(b)}$
    $w_{b+1,i}^{(k)} \leftarrow w_{b,i}^{(k)} - \eta g_k^{(b)}$
  **end for**
**end for**
$w_{t+1}^{(k)} = w_{B,I}^{(k)}$
$w_{t+1}^{(k)^*} = w_{t+1}^{(k)} + Lap(0, S^2/\varepsilon)$
$w' = w_{t+1}^{(k)^*}/q$

**Output:** $w$

$$w_{t+1}^{(k)^*} = w_{t+1}^{(k)} + Lap(0, S^2/\varepsilon). \tag{4}$$

*Step* 3: Using the AWDP method. After adding DP noise to perturb, we changed the weight attached to each participant according to (5), where $q = \sqrt{||\eta||}$. If the noisy participants are given a smaller update weight, the signal-to-noise ratio of the aggregate gradient will be better than the average method.

$$w' = w_{t+1}^{(k)^*}/q. \tag{5}$$

*Step* 4: Verification nodes use the verification incentive mechanism based on the model quality to verify the quality of the training model of the participants. If the verification is passed, the model parameters are aggregated and averaged according to (6), and the aggregated model parameters are uploaded to the blockchain as a transaction.

$$\bar{w_{t+1}} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w'. \tag{6}$$

**Theorem 1.** *SFA algorithm is effective and correct. It is resistant to the single point of failure, membership inference, and model inference attacks. It's time complexity is* $O(T \times I \times B)$.

*Proof.* The algorithm introduces blockchain technology to effectively resist single point of failure and member inference attacks. The improved DP method prevents the leakage of model parameters and does not reveal any participant information to honest but curious nodes. Suppose a total of $T$ rounds of training are to be performed. For each participant, the number of calculations required for local model training is proportional to $I \times B$. Since all participants perform model training in parallel, the time complexity is $O(T \times I \times B)$. In summary, SFA algorithm is effective and correct.

## 5 Security Analysis and Performance Evaluation

### 5.1 Security Analysis

We perform a detailed security analysis of the secure MDS model and the designed VIM and AWDP methods as follows:

1) Resistant to single point of failure attacks: Using a consortium blockchain instead of a central server in FL avoids server central server failures or mischief and helps to obtain a correct and non-toxic global model. Thus, it can prevent possible single point of failure attacks.

2) Anti-model inference attack: Using secure MDS model and AWDP method, we avoid malicious participants from inferring the original information by querying the model parameters, solve the problem of model parameters being overwhelmed by accumulated noise, and effectively protect user privacy. Thus, the prevention of model inference attacks is satisfied.

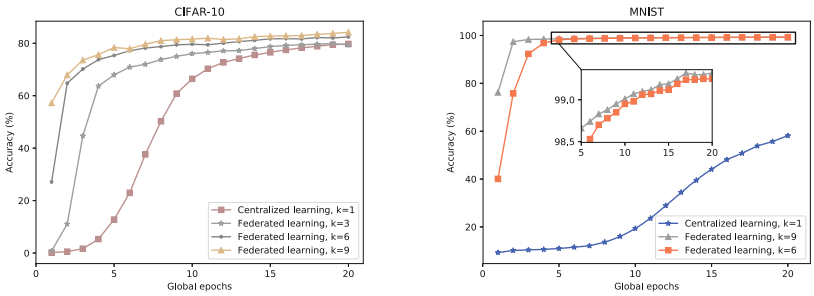
3) Resistance to Byzantine attacks and data poisoning attacks: Using VIM to validate the model quality, avoiding the possibility of unreliable participants providing false or low-quality local models, ensuring high accuracy of the model, and achieving resistance to Byzantine attacks and data poisoning attacks.

### 5.2 Performance Evaluation

In this section, the experimental setup of this study will be described, and the performance of our method will be evaluated and analyzed. We use two important indicators to measure the performance of the algorithm, namely accuracy and cross-entropy loss function. For the learning evaluation part, we evaluate two well-researched image data sets, including CIFAR-10 and MNIST, which are widely used for data classification.

As shown in Fig. 2, we compare the accuracy of federated training and centralized training on the two data sets. It can be seen that the model effect of federated training is better than that of single-point training. For single-point training on CIFAR-10, there is a big gap between the global epoch and the federated training when the global epoch is less than 15. For MNIST, even if the

global epoch reaches 20, the model accuracy is still very low. This shows that only through the data of a single client, the global distribution characteristics of the data cannot be learned well, and the generalization ability of the model is poor. In addition, the number of participants participating in the federation training in each round is different, and its performance will also have a certain difference. The greater the number of participants participating in each round of training, the higher the accuracy of the model.




(a) CIFAR-10 (b) MNIST

**Fig. 2.** Accuracy comparison of single-point training and federated training models.

In Fig. 3, we compare the accuracy changes of the model without adding differential privacy noise, simply adding differential privacy noise, adding differential privacy noise, and changing the weights in FL. The experimental results show that by assigning different weights to the size of the noise, the model parameters can be prevented from being overwhelmed by the noise. In addition, the accuracy difference between using this method and the model without adding DP is small, and our method can prevent malicious actors from malicious attacks on the local model, which improves the data security while ensuring the model's accuracy.

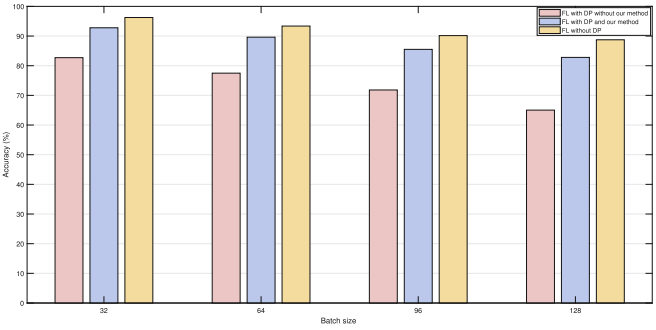



**Fig. 3.** Comparison of model accuracy in different methods.

Additionally, we evaluate the model accuracy impact of unreliable actors on federated learning tasks with different levels of data quality, namely EMD (the lower the data quality, the greater the EMD) and attack strength. From Fig. 4, we can observe that an increase in attack strength or an increase in EMD leads to a decrease in model accuracy. Therefore, unreliable users with low-quality training data or potentially malicious attacks can negatively impact model accuracy. The experimental results show that using the VIM we designed, selecting reliable participants, and validating the model quality can better improve the model accuracy.

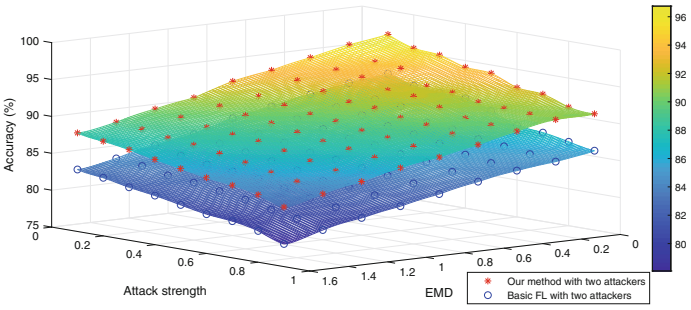



**Fig. 4.** Comparison of our method and base FL with two attackers.

## 6 Conclusions

In this paper, we built a data security sharing framework by using the consortium blockchain instead of the central server to decentralize the system and avoid the risk of a single point of failure. Second, we designed a validation incentive mechanism based on model quality. This mechanism was used to verify the model parameters before uploading them to the blockchain, thereby preventing malicious attacks and improving the enthusiasm of participants. Then, to further protect data privacy from leakage, we improved the traditional federated averaging algorithm. We added the VIM proposed in this paper and the AWDP method to the algorithm. To prevent model accuracy from being overwhelmed by noise, we assigned different weights to participants for different noises. Finally, security analysis and performance evaluation demonstrated that our method was superior in improving model accuracy and protecting user privacy. Due to the limited time, our scheme ignored time consumption, so one future direction is to reduce time consumption while improving model accuracy and data security.

**Acknowledgements.** This work was supported by the National Natural Science Foundation of China (61771289, 61832012), the Natural Science Foundation of Shandong Province with Grants ZR2021QF050, ZR2021MF075, Shandong Natural Science Foundation Major Basic Research (ZR2019ZD10), Shandong Key Research and Development Program (2019GGX1050), and Shandong Major Agricultural Application Technology Innovation Project (SD2019NJ007).

## References


1. Tang, W., Ren, J., Deng, K., Zhang, Y.: Secure data aggregation of lightweight e-healthcare IOT devices with fair incentives. IEEE Internet Things J. **6**(5), 8714–8726 (2019)
2. Liu, G., Wang, C., Ma, X., Yang, Y.: Keep your data locally: federated-learning-based data privacy preservation in edge computing. IEEE Netw. **35**(2), 60–66 (2021)
3. Qi, S., Lu, Y., Zheng, Y., Li, Y., Chen, X.: Cpds: enabling compressed and private data sharing for industrial internet of things over blockchain. IEEE Trans. Industr. Inf. **17**(4), 2376–2387 (2021)
4. Khan, L.U., Saad, W., Han, Z., Hossain, E., Hong, C.S.: Federated learning for internet of things: recent advances, taxonomy, and open challenges. IEEE Commun. Surv. Tutorials **23**(3), 1759–1799 (2021)
5. Zheng, X., Tian, L., Cai, Z.: A fair and rational data sharing strategy towards two-stage industrial internet of things. IEEE Trans. Industr. Inform. 1 (2022)
6. Cai, Z., Zheng, X., Wang, J., He, Z.: Private data trading towards range counting queries in internet of things. IEEE Trans. Mob. Comput. 1 (2022)
7. Yang, H., Zhao, J., Xiong, Z., Lam, K.-Y., Sun, S., Xiao, L.: Privacy-preserving federated learning for UAV-enabled networks: learning-based joint scheduling and resource management. IEEE J. Sel. Areas Commun. **39**(10), 3144–3159 (2021)
8. Zhou, X., Liang, W., She, J., Yan, Z., Wang, K.I.-K.: Two-layer federated learning with heterogeneous model aggregation for 6G supported internet of vehicles. IEEE Trans. Veh. Technol. **70**(6), 5308–5317 (2021)
9. Zheng, X., Cai, Z.: Privacy-preserved data sharing towards multiple parties in industrial IoTs. IEEE J. Sel. Areas Commun. **38**(5), 968–979 (2020)
10. Zhao, Y., et al.: Privacy-preserving blockchain-based federated learning for IoT devices. IEEE Internet Things J. **8**(3), 1817–1829 (2021)
11. Liu, Y., Peng, J., Kang, J., Iliyasu, A.M., Niyato, D., El-Latif, A.A.A.: A secure federated learning framework for 5G networks. IEEE Wirel. Commun. **27**(4), 24–31 (2020)
12. Jiang, C., Xu, C., Zhang, Y.: PPFL: privacy-preserving federated learning with membership proof. Inf. Sci. **576**, 288–311 (2021)
13. Cai, Z., He, Z.: Trading private range counting over big IoT data. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 144–153 (2019)
14. Gao, G., Xiao, M., Wu, J., Huang, L., Hu, C.: Truthful incentive mechanism for nondeterministic crowdsensing with vehicles. IEEE Trans. Mob. Comput. **17**(12), 2982–2997 (2018)
15. Zhao, B., Tang, S., Liu, X., Zhang, X.: Pace: privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing. IEEE Trans. Mob. Comput. **20**(5), 1924–1939 (2021)
16. Chen, W., Chen, Y., Chen, X., Zheng, Z.: Toward secure data sharing for the IoV: a quality-driven incentive mechanism with on-chain and off-chain guarantees. IEEE Internet Things J. **7**(3), 1625–1640 (2020)

# A Smart Contract-Based Intelligent Traffic Adaptive Signal Control Scheme

Wenyue Wang[1], Xiang Tian[1,2,3], Xiaolu Cheng[1,2,3], Yuan Yuan[1,2,3], Biwei Yan[1,2,3(✉)], and Jiguo Yu[2,3]

[1] School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, People's Republic of China
for_yanbiwei@163.com
[2] Big Data Institute, Qilu University of Technology, Jinan 250353, People's Republic of China
jiguoyu@sina.com
[3] Shandong Fundamental Research Center for Computer Science, Jinan 250300, People's Republic of China

**Abstract.** Intelligent traffic is one of the most important applications for improving urban traffic pressure. However, intersections are an important element of urban road network, which makes the complex traffic data face the challenges of security and efficiency in the process of transmission. In this paper, we propose a smart contract-based intelligent traffic adaptive signal control scheme to optimize the traffic efficiency problem at intersections. In the scheme, we use consortium blockchain and smart contracts to ensure secure transmission of traffic data and trusted access permission verification for intelligent traffic devices. Then, we introduce edge computing into the intelligent traffic, which can process massive traffic data in real time. In addition, we propose an improved Webster algorithm, aiming at optimizing the dynamic allocation of signal times, so as to reduce the congestion at intersections. The security analysis and evaluation experiments demonstrate that the scheme is feasible and valid, and it can facilitate the adaptive control of traffic signal lights.

**Keywords:** Intelligent traffic · Access control · Edge computing · Consortium blockchain · Smart contract

## 1 Introduction

The traffic accidents, traffic efficiency and intersections are closely related [1]. Adaptive signal control can adjust the signal lights in real time according to the actual traffic flow. Therefore, dynamic adjustment of signal lights is critical

This work was supported in part by the NSF of China under Grants 61832012 and 61771289, and the Pilot Project for Integrated Innovation of Science, The Piloting Fundamental Research Program for the Integration of Scientific Research, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant 2022XD001.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 52–63, 2022.
https://doi.org/10.1007/978-3-031-19208-1_5

to alleviate the problem of intersection congestion [2]. The vehicle network is a highly dynamic topology network, and it faces network delay of data transmission at complex traffic intersections. At the same time, malicious users may conduct malicious attacks on data and disrupt traffic order. Edge computing can solve the delay problem in intelligent traffic [3]. By deploying an edge server at the edge, the vehicle can submit tasks to the nearest edge server to process data and computing tasks. It avoids the remote transmission of data to the computing center [4]. However, managing distributed edge servers is still a challenge. Due to the influence of coverage and complex environment, the security problems of edge server are constantly exposed, such as data tampering and leakage, which cannot guarantee the security and privacy of data [5]. As a subversive innovative technology, blockchain is introduced into vehicle network, which has the characteristics of tamper-resistant and anonymity [6]. It can solve the problem of secure transmission of vehicle information and ensure the privacy of vehicle information.

In this paper, we propose a smart contract-based adaptive signal control scheme for intelligent traffic to ensure efficient processing and access control of traffic data as well as realize the dynamic control of traffic lights. The contributions of this paper are as follows.

1) We establish a distributed edge node network based on consortium blockchain and edge servers, which avoids attacks on central nodes and information asymmetry at the same time. Since edge servers such as Roadside Units (RSUs) are more robust, they are difficult to be attacked or destroyed.
2) We improve the Webster algorithm and introduce the queue length of vehicles as a key factor into the algorithm. The algorithm alleviates intersection congestion at intersections by optimizing the actual green light time of each phase.
3) We design two smart contracts to implement adaptive control of signal lights, which ensures that vehicles can securely and efficiently pass through intersections.

The rest of the paper is organized as follows. Section 2 outlines some related work. Section 3 presents the scenario problem and system model. Section 4 provides simulation results to demonstrate the superiority of the proposed scheme. Finally, Sect. 5 concludes the paper.

## 2 Related Work

### 2.1 Edge Computing

Unlike traditional cloud-based vehicle networks, edge computing allows devices to submit their tasks to “closer" edge servers [7]. Many works have considered the security and privacy of edge servers in edge computing to ensure that the data in the server will not be attacked or leaked. Cui et al. [8] proposed a VANET data download scheme based on edge computing, which allows the RSU to carry

the capability of computing and storing data, improving the efficiency of vehicle data requests. Wang et al. [9] proposed a triple real-time trajectory privacy protection mechanism (T-LGEB) in order to protect the trajectory privacy of task participants while ensuring the real-time nature of the data.

### 2.2 Consortium Blockchain

The consortium blockchain is a type of blockchain, which can pre-select a certain number of accounting nodes to verify the validity of data and blocks. Therefore, it takes less time to reach a consensus [10]. Smart contracts are scripts that are executed automatically and independently, which are triggered by events [11]. Today, there have been many studies in the academic community to solve related problems in edge computing by combining consortium blockchain. In [12], Rivera et al. proposed a consortium blockchain-based task sharing framework, which allows vehicles to enable task offloading and sharing among edge servers for efficient cooperation.

### 2.3 Signal Timing Optimization

Webster's method is the most classic intelligent traffic timing algorithm, which takes the delay time of vehicles passing through the intersection as the only metric, and then optimizes the signal timing scheme [13]. Subsequently, scholars have also achieved a lot of results in the research on signal timing. In [14], Adeke et al. used the Webster algorithm to set different static timing schemes according to the traffic conditions in each specific time period. Tajalli et al. [15] coordinate vehicle speed and reduce the number of stops at signalized intersections to smooth traffic flow. Liang et al. [16] dynamically determined the duration of the signal light by collecting traffic data from different sensors, instead of dividing the signal period into multiple segments.

## 3 Intelligent Traffic Signal Control Solution

### 3.1 Problem Description

Figure 1 shows a simple example. The Sun Road is green light, and the Wit Road is red light. When the queue of vehicles waiting for the red light on Wit Road is very long, there are few passing vehicles on Sun Road, but there is still 30 s left in the green time. At this time, vehicles waiting on the Wit Road can be regarded as invalid waiting, and the traffic efficiency has the potential to be improved.

A possible solution to the above problem is to first calculate the time for the remaining vehicles to pass the signal light within a fixed range on Sun Road (50 m, ignoring the continuous traffic coming from behind), and then reduce the green time of Sun Road. Meanwhile, we should also correspondingly increase the green time of Wit Road, and shorten the waiting time of some vehicles on Wit Road.

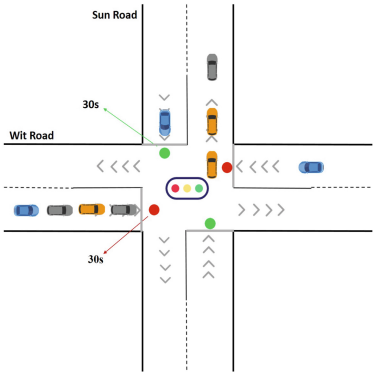



**Fig. 1.** Problem description.

### 3.2 System Design

As shown in Fig. 2, the system consists of three parts: Data layer, edge layer and cloud layer. In the data layer, signal lights and cameras interact with RSU. In the edge layer, four RSUs (edge nodes) deployed at each intersection are combined into an edge cluster. Edge clusters temporarily store and process simple and time-sensitive tasks, and transmit data to the cloud layer through wired connections when necessary. In the cloud layer, the central authority (CA) manages all edge clusters, which can permanently store massive amounts of data and perform complex and latency-tolerant tasks for the edge layer.

### 3.3 System Model

Figure 2 shows the system model of the signal control scheme in intelligent traffic. The system architecture can be divided into the following different components, which are described below.

1) **Central Authority(CA)**. During system initialization, CA is responsible for the identity registration and key distribution of RSU, cameras and signal lights.
2) **Roadside Unit(RSU)**. We use RSUs as edge devices (nodes), which are mainly responsible for processing and temporarily storing data uploaded by intelligent traffic devices. Moreover, it can also dynamically control traffic lights through smart contracts. Each RSU manages a pair of transportation equipments(camera and signal light) at its intersection.
3) **Camera**. The cameras are placed above the signal lights, and the queue lengths of vehicles in the lane is calculated periodically and the results are sent to the RSU to which they belong.
4) **Signal light**. The signal light periodically takes their own signal status and time, and then it sends them to their own RSUs.

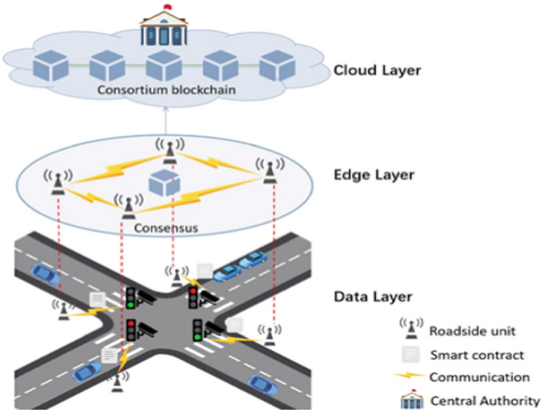



**Fig. 2.** Structure of the system model.

5) **Information Management Contract (IMC)**. The main purpose of IMC is to manage the access control permission of transportation equipments. There are two lists for IMC as follows:
   - **Permission list**. When initializing the list, pre-defined policies will be used for transportation equipments to realize static access permission verification. As shown in Table 1, we record the ownership and access permissions of IoT devices.
   - **Behavior list**. It records the misbehaviors and corresponding punishments by transportation equipments during the communication process as shown in Table 2.

6) **Judgment Contract (JC)**. It is mainly to judge whether the red light lane satisfies the conditions for reducing the invalid waiting time of vehicles. RSU receives the judgment result output by JC, and dynamically adjusts the signal light period according to the result.

The basic fields for each row of Table 1 are shown below:

- Object: The RSU is responsible for managing a pair of signal light and camera.
- Subject: Signal lights and cameras. The access policy is limited to the subject which sends data to the object it belongs to.
- Permission: When the list is initialized, all permissions are allowed. Once the object misbehaves, the permissions will be changed to deny.

The basic fields for each row in Table 2 are shown below:

- Object: The subject of misbehavior.
- Misbehavior: Misbehavior by the subject.
- Time: The time when the subject committed the misbehaviors.
- Penalty: Revoking the access permission of the object within a certain period of time.

**Table 1.** Permissions of list.

| Subject | Object | Permissions |
|---|---|---|
| $rsu_1$ | $Signal_1$ | Allow |
| | $Camera_1$ | Allow |
| $rsu_2$ | $Signal_2$ | Allow |
| | $Camera_2$ | Deny |
| ... | ... | ... |

**Table 2.** Misbehaviors of list.

| Subject | Misbehavior | Time | Penalty |
|---|---|---|---|
| $Camera_2$ | Send the wrong length of the vehicle queue | 2022-3-15 21:00 | locked for 3 h |
| ... | ... | ... | ... |

### 3.4 Scheme Flow

Figure 3 shows the processes of model. Next, we will introduce the steps of the proposed scheme.

**Step 1: System Initialization and Key Generation.** We initialize the system by using elliptic curve digital signature algorithm and asymmetric encryption. After each signal light, camera and RSU are authenticated by the CA, the CA will generate an ID and a pair of public/private keys for them (i.e. $ID_{u_i}$, $PK_{u_i}$, $SK_{u_i}$, $Cert_{u_i}$,$u$ represents a certain intelligent traffic devices). Both the ID and the public/private key pair (PK, SK) will be used as the identity of the IoT, which can be double-authenticated to ensure the authenticity of the device identity.

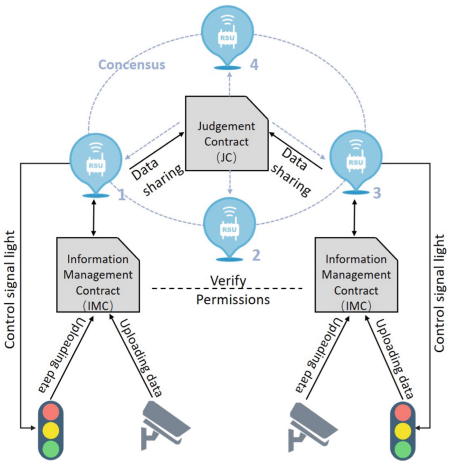



**Fig. 3.** The processes of mode.

**Step 2: Uploading Data.** Signal lights and cameras upload data to the RSU regularly. First, it sends a request to the RSU to which it belongs, and obtains the symmetric key $K$, which is used to encrypt the uploaded data, increasing the data transmission efficiency. The process of uploading data is divided into the following two parts: authorization verification, key distribution and data encryption.

*Authorization Verification.* Signal lights and cameras upload requests to $\mathrm{RSU}_i$ through IMC. First, the IMC verifies the relationship between the signal light and the RSU through the Permission list. If the $\mathrm{ID}_{r_i}$ and $\mathrm{ID}_{s_i}$($r$, $s$ represent RSU and signal light, respectively) correspond, then the signal light has permission to access the RSU. If not, the request will be recorded as a misbehavior in the Behavior list. Then, the IMC checks the Behavior list for penalties. If not, the $\mathrm{RSU}_i$ receives the request. At this point, the signal light has completed the first authorization.

*Key Distribution and Data Encryption.* $\mathrm{RSU}_i$ will perform the second authentication on the signature $\mathrm{Sig}_{s_i}$ and the certificate $\mathrm{Cert}_{s_i}$ of the signal light $\mathrm{S}_i$. If the authentication is passed, the symmetric key $K_s$ is sent to the signal light $\mathrm{S}_i$. In order to ensure the secure transmission of data, the generation and transmission of symmetric keys are performed periodically at regular intervals and are distributed to the managed devices by $\mathrm{RSU}_i$.

$$\begin{aligned} &s_i \rightarrow rsu_i: \ \text{Request1} \ = \{ID_{r_i}, ID_{s_i}, M_1\} \\ &where \\ &M_1 = \mathrm{Enc}_{PK_{r_i}}\left(Cert_{s_i}, Sig_{SK_{s_i}}\left(Nonce_a, Timestamp_1\right)\right). \end{aligned} \tag{1}$$

$$\begin{aligned} &rsu_i \rightarrow s_i: \ \text{Reply1} \ = \{ID_{r_i}, ID_{s_i}, M_2\} \\ &where \\ &M_2 = \mathrm{Enc}_{PK_{s_i}}\left(Cert_{r_i}, Sig_{SK_{r_i}}\left(K_s, Nonce_b, Timestamp_2\right)\right). \end{aligned} \tag{2}$$

$$\begin{aligned} &s_i \rightarrow rsu_i: \ \text{Reply2} \ = \{ID_{s_i}, ID_{r_i}, M_3\} \\ &where \\ &M_3 = \mathrm{Enc}_{K_{s_i}}\left(Sig_{SK_{s_i}}\left(Nonce_b, Timestamp_3\right)\right). \end{aligned} \tag{3}$$

$$\begin{aligned} &s_i \rightarrow rsu_i: \ \text{Upload} \ = \{sID_i, rID_i, M_4\} \\ &where \\ &M_4 = \mathrm{Enc}_{K_s}\left(Data, Timestamp_4\right). \end{aligned} \tag{4}$$

As in (1), the Request1 indicates that the signal light $\mathrm{S}_i$ sends a request to $\mathrm{RSU}_i$ for uploading data. The message $M_1$ indicates the uploaded data request certificate, which is encrypted by the public key PK of $\mathrm{RSU}_i$. $\mathrm{Sig}_{SK_{s_i}}$ is signed by the private key of $\mathrm{S}_i$, ensuring that messages cannot be tampered with. $M_2$ is encrypted by the public key of $\mathrm{S}_i$. The symmetric key $K_s$ is encrypted in the

signature by $\mathrm{RSU}_i$ with the private key SK. As in (2), If $\mathrm{S}_i$ can decrypt $M_2$ correctly, it will obtain the key $K_s$ and send a receive response Reply2 to $\mathrm{RSU}_i$. As in (3), $M_3$ is encrypted with the key $K_s$.Upload represents the data uploaded by $\mathrm{S}_i$. As in (4), $M_4$ is encrypted by the symmetric key $K_s$, and *Data* represents the uploaded data content.

The key distribution and data encryption process of the camera is the same as the signal light $\mathrm{S}_i$.

**Step 3: Data Sharing.** $\mathrm{RSU}_i$ decrypts the received encrypted data $\mathrm{Dec}_{K_{s_i}}(M_4)$. Then, the obtained data is judged by JC and sent to the edge cluster to execute the consensus process.

*JC Execution Judgment.* First, JC judges whether the queue length of vehicles on the red light lane reaches the threshold. Then, JC calculates the time *Vtime* required for the remaining vehicles in the green light lane to pass through the intersection, and compares whether the difference between the signal light and *Vtime* is valid ($\geq 5$ s, considering vehicle start-stop and network delay). If the difference is valid, JC uploads the difference to the edge cluster and returns it to the RSU to dynamically control the signal light.

**Algorithm 1:** Information Management Contract (IMC)

**input :** sID, cID, rID
**output:** $M_4$

**if** *sID,cID belongs to rID* **then**
  **if** $rights_{sID} = allow$ *and* $rights_{cID} = allow$ **then** *uploaded data* $\leftarrow$ *true*;
   `// Both rights must be allowed`
  **else** *uploaded data* $\leftarrow$ *false*;
**else**
  *rights* = *deny*;
  `// Modify the rights to deny`
  *penalty* = *locked for 3 hours*;
  `// Give it penalty based on the current time CUtime`
**end**

The traditional Webster algorithm takes the minimum total delay time of vehicles as an indicator, and does not consider the queue length of vehicles. Dynamically allocating the green time of signal lights can reduce the waiting time and start-stop times of vehicles and improve traffic efficiency. Therefore, in our improved Webster algorithm, the queue length of vehicles is introduced as one of the indicators to obtain the optimal signal period.

According to the Webster algorithm, the scheme of improving the phase setting is used to calculate the optimal period length of the intersection [13].

**Algorithm 2:** Judgement Contract(JC)

**input :** $Q_i$, $Q_j$, $Ge_i$
**output:** $Ge_j$ or null

*The subscripts i and j represent the green and red lanes, respectively*;
**if** $Ge_j \geq 50$ **then**
  $Vtime = Q_i \ / \ v_c$;
  **if** $Vtime < Ge_i$ **then** `// Time for vehicles in the green lane to pass the intersection`
    **if** *green light lane with cars* **then** $Ge_j = Ge_i - Vime$;
    `// Ge`$_j$ `is for time in the red lane`
    **else** $Ge_j = Ge_i$;
  **end**
  $Ge_i = 0$;
  `// Green light lanes change to red lights`
**end**

$$C_0 = \frac{1.5\,L + 5}{1 - Y} \tag{5}$$

In (5), $C_0$ is the optimal signal period, $L$ is the total loss time.

$$Y = \sum_{i=1}^{n} y_i, \quad y = \frac{q}{s} \tag{6}$$

In (6), $Y$ is the traffic flow ratio of the intersection. $y$ is the phase critical flow ratio, $q$ is the lane demand, and $s$ is the lane saturation flow.

$$L = \sum_{i=0}^{n} (l_i + l + A) \tag{7}$$

In (7), $l_i$ is the loss time of the vehicle starting, which is generally 3 s. $A$ is the yellow light time that is about generally 3 s. $l$ is the green light interval time, $n$ is the number of signal phases.

$$Ge_i = \frac{y_i}{Y}\,(C_0 - l) \tag{8}$$

$C_0 - l$ is the total of the effective green time calculated in (8).

$$Q = \frac{q}{3600/c} \tag{9}$$

In (9), $Q$ is the queue length of the vehicle, $q$ is the lane demand flow, and $c$ is the cycle length.

$$Ge_j = \frac{y_i}{Y}\,(C_0 - l) - \frac{Q_i}{V_c} \tag{10}$$

In (10), $Ge_j$ is the signal light time on the red light lane, $Q_i$ is the queue length of the vehicle on the green light lane, and $V_c$ is the speed of the vehicle passing through the intersection. *A* modified Webster algorithm for the above analysis is described in JC.

**Step 4: Generate Blocks.** Each RSU in the edge cluster collects the data uploaded into the edge cluster. RSU generates a new data block with a timestamp and broadcasts it to other RSUs in the consortium blockchain for verification and audit.

## 4 Security Analysis and Experimental Results

### 4.1 Security Analysis

The consortium blockchain ensures the traceability of data, and the automatic execution of smart contracts ensures trusted access control of the system. We analyze the security performance of our scheme as follows:

1) **Fine-grained access control**. Permission list and smart contract ensure the subject-to-object access control and the credibility of data sources.
2) **Credibility of uploaded data**. When the edge node decrypts the encrypted data, the JC contract is triggered immediately, and then the data is uploaded to the edge node cluster for consensus. Therefore, it is difficult for malicious nodes to find out when the data was tampered with.
3) **Shared data authentication**. All shared data is publicly audited and authenticated by other nodes. It is impossible to compromise all nodes due to overwhelming cost. Therefore, faulty shared data can still be discovered before the block is built.

### 4.2 Experimental Results

Figure 4 shows the relationship between smart contract and the times of uploaded data of intelligent traffic devices. We found that with the increase in the number of uploads from cameras and signal lights, the runtime cost of smart contracts gradually stabilized.

**Table 3.** Average time consumed by each process.

| Process | Time consumed (ms) |
|---|---|
| Verify rights (MC) | 1.24 |
| Data encryption | 91.3 |
| Data decryption | 139.35 |
| Judge conditions (JC) | 0.97 |

The whole process of uploading data is divided into four parts, and the average time consumed by each part is shown in Table 3. The average time for traffic lights and cameras to upload encrypted data is 91.3 ms, and the encryption time is proportional to the size of the uploaded data. The average time for IMC to verify access was 1.24 ms. Due to the delay of network environment, the time cost of smart contract should be slightly higher than the average time cost of experiment in real situation. The average time for RSU to decrypt data is 139.35 ms. The average time for JC to make a decision is 0.97 ms. And the total time consumption of the whole process is about 233 ms.

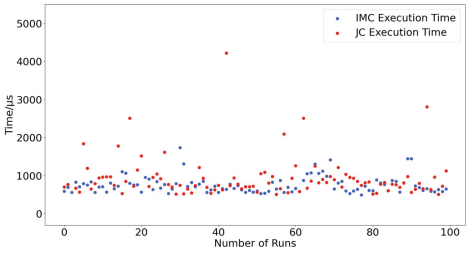



**Fig. 4.** Smart contract runtime.

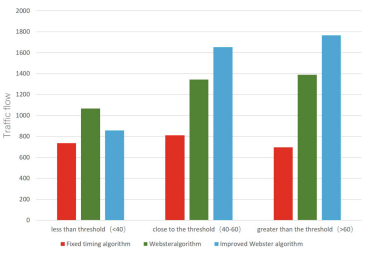



**Fig. 5.** Changes in traffic flow.

As shown in Fig. 5, under the control of three different algorithms, we compare the total traffic flow at the intersection when the queue length of vehicles is less than the threshold(<40 m), close to the threshold(40-60 m) and greater than the threshold(>60 m), respectively. In order to ensure traffic security, the speed of urban traffic should not exceed 60 km/h. Within 120 min, we set 200 vehicles to pass through the intersection in turn. In the case of less than the threshold, the effect of the improved Webster algorithm is very close to that of the fixed timing method. When the total traffic flow is close to the threshold and greater than the threshold, the improved Webster algorithm has the highest traffic flow through the intersection, followed by the Webster algorithm. Therefore, the improved Webster algorithm is more suitable for the situation where there are many vehicles at the intersection, which can significantly improve the traffic efficiency.

## 5 Conclusion

In this paper, a smart contract-based adaptive signal control scheme for intelligent traffic is proposed. We use consortium blockchain and smart contract technology to achieve trusted access control between intelligent traffic devices, effectively preventing unauthorized and untrusted data communication. Then, we introduce edge computing into the model, which greatly optimizes the calculation and storage of massive traffic data in the intelligent traffic. Finally, we

propose improved Webster algorithm, which reduces the influence of the queue length of vehicles on intersection.

## References


1. Poch, M., Mannering, F.: Negative binomial analysis of intersection-accident frequencies. J. Transp. Eng. **122**(2), 105–113 (1996)
2. Wang, T., Cao, J., Hussain, A.: Adaptive traffic signal control for large-scale scenario with cooperative group-based multi-agent reinforcement learning. Transp. Res. Part C: Emerg. Technol. **125**, 103046 (2021)
3. Cui, L., Yang, S., Chen, Z., Pan, Y., Ming, Z., Xu, M.: A decentralized and trusted edge computing platform for internet of things. IEEE Internet Things J. **7**(5), 3910–3922 (2019)
4. Firdaus, M., Rhee, K.H.: On blockchain-enhanced secure data storage and sharing in vehicular edge computing networks. Appl. Sci. **11**(1), 414 (2021)
5. Manivannan, D., Moni, S.S., Zeadally, S.: Secure authentication and privacy-preserving techniques in vehicular ad-hoc networks (vanets). Veh. Commun. **25**, 100247 (2020)
6. Maesa, D.D.F., Mori, P.: Blockchain 3.0 applications survey. J. Parallel Distrib. Comput. **138**, 99–114 (2020)
7. Khan, L.U., Yaqoob, I., Tran, N.H., Kazmi, S.A., Dang, T.N., Hong, C.S.: Edge-computing-enabled smart cities: a comprehensive survey. IEEE Internet Things J. **7**(10), 10200–10232 (2020)
8. Cui, J., Wei, L., Zhong, H., Zhang, J., Xu, Y., Liu, L.: Edge computing in vanets-an efficient and privacy-preserving cooperative downloading scheme. IEEE J. Sel. Areas Commun. **38**(6), 1191–1204 (2020)
9. Zhang, J., Zhong, H., Cui, J., Tian, M., Xu, Y., Liu, L.: Edge computing-based privacy-preserving authentication framework and protocol for 5g-enabled vehicular networks. IEEE Trans. Veh. Technol. **69**(7), 7940–7954 (2020)
10. Wang, S., Ye, D., Huang, X., Yu, R., Wang, Y., Zhang, Y.: Consortium blockchain for secure resource sharing in vehicular edge computing: a contract-based approach. IEEE Trans. Netw. Sci. Eng. **8**(2), 1189–1201 (2020)
11. Hewa, T., Ylianttila, M., Liyanage, M.: Survey on blockchain based smart contracts: applications, opportunities and challenges. J. Netw. Comput. Appl. **177**, 102857 (2021)
12. Rivera, A.V., Refaey, A., Hossain, E.: A blockchain framework for secure task sharing in multi-access edge computing. IEEE Netw. **35**(3), 176–183 (2020)
13. Webster, F.V.: Traffic signal settings. Tech. rep. (1958)
14. Adeke, P.T., Atoo, A.A., Zava, A.E.: Traffic signal design and performance assessment of 4-leg intersections using webster's model: a case of 'srs'and 'b-division'intersections in makurdi town. Int. Res. J. Eng. Technol. **5**(5), 1253–1260 (2018)
15. Tajalli, M., Mehrabipour, M., Hajbabaie, A.: Network-level coordinated speed optimization and traffic light control for connected and automated vehicles. IEEE Trans. Intell. Transp. Syst. **22**(11), 6748–6759 (2020)
16. Liang, X., Du, X., Wang, G., Han, Z.: A deep reinforcement learning network for traffic light cycle control. IEEE Trans. Veh. Technol. **68**(2), 1243–1253 (2019)

# Inferring Device Interactions for Attack Path Discovery in Smart Home IoT

Mengjie Sun[1,2], Ke Li[1,2], Yaowen Zheng[3], Weidong Zhang[1,2], Hong Li[1,2], and Limin Sun[1,2(✉)]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{sunmengjie,like,zhangweidong,lihong,sunlimin}@iie.ac.cn
[3] Nanyang Technological University, Singapore, Singapore
yaowen.zheng@ntu.edu.sg

**Abstract.** In smart home Internet of Things (IoT) systems, interactions between devices are driven in two ways: automation applications (apps), and the physical channels (e.g., temperature, smoke). Meanwhile, device interactions can be maliciously exploited to launch IoT attacks. However, limited efforts explore whether it is feasible to discover potential exploitable device interactions from IoT deployments. This paper proposes a novel framework to **d**etect **interact**ions among devices (D-interact) from eavesdropped network traffic and device function descriptions, and discover all potential exploitable device interactions (i.t., attack paths). First, we use the packet-level patterns to fingerprint IoT device events and then identify all IoT device events from the eavesdropped traffic. Furthermore, we mine temporal and conditional dependencies of IoT events to infer device interactions introduced by IoT apps. Besides, to identify interactions between devices and physical channels, we use the natural language processing (NLP) technique to analyze device function descriptions. Based on the obtained device interactions, D-interact builds a device interaction graph to discover attack paths. To demonstrate the feasibility of our approach, we implement D-interact in a real-world smart home including 24 devices and 29 apps. The experiment results show that 38 device interactions are identified and 26 device interaction paths could be potentially exploited to impact the safety of the IoT environment.

**Keywords:** Smart home · Traffic analysis · Attack path discovery · Physical interaction control

## 1 Introduction

With the rapid development of smart homes, the IoT technologies (especially IoT platform) are utilized to control the devices remotely and automatically, achieving a convenient and energy-saving living environment. Both commercial


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 64–75, 2022.
https://doi.org/10.1007/978-3-031-19208-1_6

and open-source IoT platforms for smart homes are readily available on the market, such as Samsung SmartThings [15], and HomeAssisant [2], which support home automation applications(e.t., IoT apps).

IoT technologies bring new security issues to the smart home environment. In recent years, many security problems have been revealed on both IoT devices and platforms, such as device firmware flaws [3], communication protocol flaws [1,4], information leakage [13], and malicious applications [7]. Besides, security problems caused by interactions of IoT apps/devices have recently attracted significant attention [5,6,14]. Specifically, the automated interactions of devices in an smart home can be broadly classified into two categories: 1) **app-enabled interactions**, where the control logic of an app directly connects two devices through network communication, and 2) **physical interactions**, where actuators change the physical environment through executing commands, and sensors report changes of the physical environment through events [14]. The interaction features of IoT devices can be potentially exploited by attackers to jeopardize IoT environments.

In particular, the attackers can manipulate and exploit the device interactions to exacerbate physical risks. Considering a smart home environment where an app opens a window when the temperature is above a preset threshold. To break into the house, an attacker may manipulate the temperature physical channel to maliciously trigger control logic in the app to open the window. In this paper, we call such device interactions (i.t., the manipulated temperature channel controls a thermometer to report changes in temperature, then the reported event triggers a window open command through the app) as an attack path. Such attack paths could undermine the security and safety of smart homes.

Given a real-world smart home, discovering potential exploitable device interactions (attack paths) presents two challenges. First, how to effectively infer basic smart home configurations (e.g., IoT device types, IoT apps) to identify device interactions. Second, how to construct and discover complete attack paths. Previous works [8,13,17,18] have illustrated the effectiveness of inferring smart home configuration information from network traffic. They can identify device information, IoT events/commands, and even installed IoT apps. However, these works failed to recognize devices' physical interactions and construct complete attack paths in a realistic scenario.

In this paper, we propose a novel approach called D-interact, which infers the device interactions introduced by IoT apps and physical channels and discovers all potential exploitable attack paths. To start with, we identify both app-enabled interactions and physical interactions in smart home system. For app-enabled interactions, we use the packet-level patterns to fingerprint IoT device events, then identify all IoT events and device information from constantly eavesdropped traffic. Based on discovered IoT events, we identify app-enabled interactions by mining the conditional and temporal dependencies of IoT events. For the physical interactions, we identify them by applying natural language processing (NLP) techniques to publicly available descriptions of device functions. Finally, With both app-enabled and physical interactions, we build a device interaction graph and use a depth-first-search algorithm to discover potential attack paths.

We evaluate D-interact in an real-world smart home system, which includes 24 devices and 29 automation rules and is based on IoT platform HomeAssistant. We collect traffic from 4 testbed rooms for two weeks by sniffers. D-interact identified 38 device interactions, including 17 app-enabled interactions and 21 physical interactions, and discovered 26 attack paths. The experiments demonstrate that D-interact can help attackers to infer exploitable device interaction paths to attack realistic smart home systems.

**Contributions**. Our main contributions are outlined below:

- To the best of our knowledge, it is the first work to discover potentially exploitable device interaction paths inferred from the network traffic and device function descriptions.
- We design a novel algorithm to identify devices' app-enabled interactions via mining the temporal and conditional IoT events dependencies. We propose a new mechanism to identify interactions between devices and the physical environment.
- We evaluate the proposed D-interact on a real smart home. Attackers can exploit the discovered attack paths by D-interact to make severe physical consequences.

## 2 Background: Smart Home IoT

In modern smart homes, IoT devices are integrated via IoT platforms for rich automation. IoT platforms typically include devices, connectivity protocols, and an IoT cloud backend. Connectivity protocols are various such as WiFi, Zigbee, and Z-Wave. Usually, a hub is responsible for the communication between the physical devices and the cloud backend.

**IoT Devices:** IoT devices are classified as actuators and sensors and each device has its associated events and commands. An event is defined as the status change of a stand-alone smart device, while a command is generated by the cloud backends and can control actuators. An actuator (e.g., plug) can receive and execute commands (e.g., switch on/off) to change its status and report an event after executing a command, while a sensor(e.g., motion sensor) measures the surrounding environment and send them directly or via a hub to the cloud.

**Automation:** The cloud backend can synchronize device states by device's latest events and run IoT apps, which follow the trigger action programming paradigm. All IoT apps are made up of a trigger event, an action command and an optional condition. An app is activated when a certain trigger event is received, and then the action is taken after confirming the condition is met by the states (e.g., device state, time) stored in the cloud.

## 3 Motivation and Threat Model

### 3.1 Motivation Examples

With the prior knowledge of device interactions, attackers can launch attacks on well-protected devices from resource-constrained devices by maliciously

triggering such interactions and undermining the security of the whole system. Assuming a smart home scenario in Fig. 1, there are three deployed devices (a heater, a thermometer, and a window opener) and an app $APP_1$ which opens a window when the temperature of a room is above a given threshold. There are two type of attacks to open the window and break in. (1) Attackers directly launch an event spoofing attack and fake a temperature-detected event with a high value on the thermometer, triggering $APP_1$ to open the window. (2) The attacker could maliciously control the compromised heater to execute the heater-on command to raise the indoor temperature, triggering $APP_1$ to open the window.

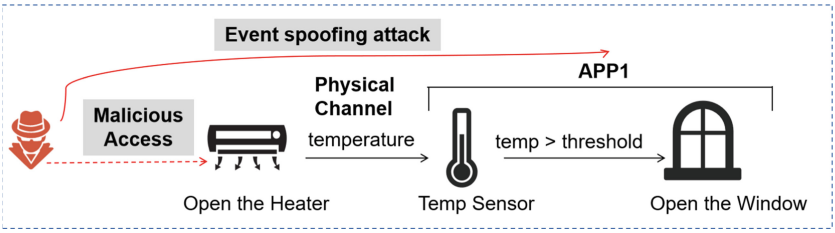



**Fig. 1.** Attackers exploit device interactions caused by IoT $APP_1$ and the temperature physical channel to automatically open the well-protected window.

### 3.2 Threat Model

The attackers aim to leverage device interactions to attack a real smart home. The capabilities of attackers are similar to [1,8,13]. Attackers can sniff the traffic of common communication protocols (e.g., wifi, Zigbee). For this, attackers need to physically deploy the sniffers and then can gather the network traffic remotely. Attackers can also access the traffic flow through cyberattacks such as weak router passwords and ARP attacks. During passively sniffing the network traffic, the attackers may not be detected by the victim for a long time.

We also assume attackers can access the same types of IoT devices as benign smart home users. This is reasonable that IoT devices are available on the market. Besides, we assume that the IoT platform software and hardware are trustworthy.

## 4 Design

As shown in Fig. 2, the design of our proposed framework includes two modules: 1) Device Interaction Identification, and 2) Attack Path Discovery.

The *Device Interaction Identification* modlue includes two methods: (1) *App-enabled Interaction Identification*, which infers the devices' app-enabled interactions by analyzing the network traffic, and (2) *Physical Interaction Identification*, which infers the physical interactions by analyzing each device's function descriptions. The *Attack Path Discovery* module takes device interactions as input to build a device interaction graph, and the depth-first-search technique is exploited to discover potential exploitable attack paths in this graph.

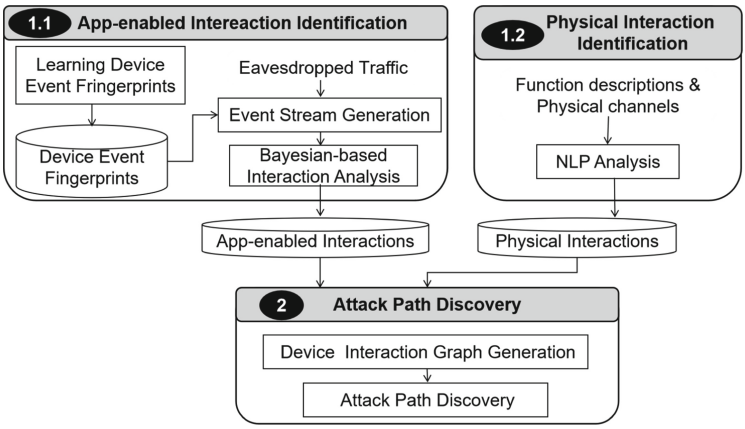



**Fig. 2.** D-interact system overview.

### 4.1 App-enabled Interaction Identification

In this section, we first describe how we learn the fingerprint of device events. Then we identify all device events and deployed device information from network traffic by the matching learned event fingerprints. Furthermore, we infer app-enabled interaction by mining the dependencies of the events.

**Learning Device Event Fingerprints.** Typically, IoT device events(e.g., light bulb turn on/off) have unique packet-level network patterns. Whenever a device event happens, a sequence of time-ordered wireless packets is generated between the device and the cloud backend. Similar to [17,18], the most frequently appeared packet sequence is identified as the fingerprint of such event. Particularly, the network fingerprint of a device event is defined as

$$F = (p_1, p_2, ..., p_n), \tag{1}$$

where $p_i$ is a packet, presented as $p = (timestamp, length, direction)$, and the interval between any two consecutive packets is less than a pre-determined threshold $\lambda$ ($\lambda$ is called the burst threshold). Some sensors report discrete values (e.g., temperature value), and typical IoT wireless communication protocols encrypt the data payload field. So we do fingerprint such events to know the occurrence of these events and do not know what the changed value is.

Specifically, in the step of Device Event Fingerprint Learning, we manually trigger each event $m$ times and collect $m$ samples, denoted as $S = \{S_1, S_2, ..., S_m\}$, where a sequence of packets $S_i$ is collected in one experiment. We use the Levenshtein distance as the measurement, and a small value indicates a high similarity. For each sequence $S_i$, we calculate $\sum_{j=1}^{m} dist(S_i, S_j)$, which is the sum of the distance between this sequence and other sequences. Then the sequence, which has the smallest sum result and the largest similarity with the other sequences, is chosen as this device event's fingerprint.

**Event Stream Generation.** After capturing the network traffic by corresponding packet sniffers, we need to identify all device events and infer the information of deployed IoT devices. Specifically, we infer the number of devices according to the unique network address in the packet header, which is usually unique and unencrypted and can be used to distinguish IoT devices. Besides, we can infer device types and their events/commands from the network traffic by matching the unique device event fingerprints.

To generate device event streams, we first group the traffic by the network address. Each of the grouped traffic is denoted as $P_{d_i} = (p_1, p_2, ..., p_n)$ for a device $d_i$, where $p_i$ is a packet. For each $P_{d_i}$, we filter out the unrelated packets, including beacon packets and retransmission packets. Then we split the traffic into bursts. A burst is a time-ordered sequence of packets and the interval between any two consecutive packets is less than the burst threshold $\lambda$. Next, each burst is matched with the fingerprint of each device event by calculating the Levenshtein Distance $l_{E_i} = dist(S_b, S_{E_i})$. If the smallest $l_{E_i}$ is 0, we identify the mapped event $E_i$. Finally, each $P_{d_i}$ is converted into a event stream for device $d_i$.

To mine the dependencies between different device events, we merge all device event streams in time-ordered sequence and construct the total event stream $ES =< (d_1, e_1, t_1), (d_2, e_2, t_1), ..., (d_k, e_k, t_k) >$, where $e_i (i = 1, ..., k)$ is the IoT device $d_i$ event happening at time $t_i$. Besides, we use the feature that a device's events typically exist in pairs (e.g., on and off, active and inactive), to distinguish the same fingerprints of the events.

**Bayesian-Based Interaction Analysis.** An app-enabled interaction has a trigger event $E_T$ and an action command $E_A$, which are device events generated by a corresponding app. The trigger event and action command frequently appear together within a short interval. Besides, when the action command appears, the trigger event of the relevant app must have appeared in the network traffic.

We use the Bayesian conditional probability method [10] to mine cyberspace interactions from the event stream $ES$. If two IoT device events interact through the same app, the time interval between events is less than the time threshold $\sigma$ ($\sigma$ is determined by automation processing time and network latency) and they have condition-dependent relationships. This method contains three steps.

1) We distinguish between trigger events and action commands. Specifically, only events of actuator devices are treated as action commands and all device events could be trigger events.

2) We get all trigger-action pairs and the number of occurrences. For each type action command $E_A$ in event stream $ES$, it has a timestamp $T_A$ and its corresponding trigger events $E_{Ti} (i \in \mathbb{N})$ occur within $\sigma$ seconds before $T_A$. Then, we count the action command $E_A$ and get $N_A$ (the number of $E_A$). For each corresponding trigger event, we get $N_{Ti}$ (the number of $E_{Ti}$) and $N_{TiA}$ (the number of corresponding trigger-action pair).

3) We calculate the conditional probability of

$$P(E_T/E_A) = N_{TA}/N_A, P(E_A/E_T) = N_{TA}/N_T. \quad (2)$$

Then the trigger-action pairs, whose neither $P(E_T/E_A)$ nor $P(E_T/E_A)$ is less than 0.9, is chosen as app-enabled interactions.

### 4.2 Physical Interaction Identification

Physical interactions depend on physical laws and the physical functions of devices. Typically, devices' physical interactions span a long and uncertain period of time and the method for inferring app-enabled interactions is unworkable. Physical functions of devices are encoded in device function descriptions.

We leverage NLP techniques to identify physical interactions between devices and the physical channels. The inputs include the collected device function descriptions and seven common physical channels, including temperature, humidity, water, smoke, illumination, motion, and sound. This method contains three steps.

1) We extract all entity keywords from device function descriptions. First, we tokenize and conduct part-of-speech (POS) tagging to the descriptions. Then we choose the nouns as entity keywords and lemmatize those nouns to remove inflectional endings and return the base or dictionary form of a word.

2) For each device, we match the physical channels to the corresponding entity keywords and then get the physical channels associated with this device.

3) For each device, we construct physical interactions by matching each actuator command and sensor event with corresponding physical channels. Specifically, actuators change the physical environment by executing actuation commands, while sensors record the change of the surrounding environment. We use pointing arrows to distinguish physical interactions, denoted as $< actuator, command >\rightarrow physical\ channel$ and $< sensor, event >\leftarrow physical\ channel$.

### 4.3 Attack Path Discovery

Based on the inferred app-enabled interactions and identified physical interactions in the above steps, our system further builds a device interaction graph to discover all potential exploitable attack paths.

The device interaction graph $G = < V, E >$ is a directed graph and contains two basic elements vertexes $V$ and edges $E$. The $V$ includes IoT device events/commands and the physical channels. The $E$ includes both app-enabled and physical interactions. The attack path is defined as a interaction path from the root node to the leaf node with a path length greater than one. By exploiting such attack paths, attackers can use vulnerable devices to maliciously trigger cascading interactions between devices and eventually affect the well-protected device (e.g., a anti-theft door is opened automatically, even if no one is at home). We discover the attack paths through two steps.

1) We build a directed graph $G$, and then add all type device events and the physical channels as vertexes and add all inferred device interactions as edges.

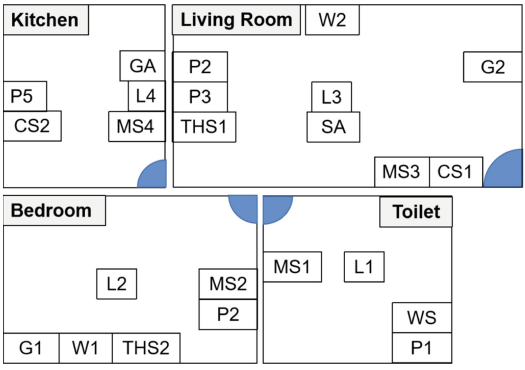
2) Then, we use the depth-first-search algorithm [16] to find paths that from the vertex without in-degree to the vertex without out-degree. Paths whose

lengths are greater than one in the graph are considered as potential attack paths, since such paths associate two devices and can be exploited by attackers.

## 5 Experiments and Results

### 5.1 Experimental Setup

**Testbeds, Participants and IoT Apps.** We deploy a real smart home environment based on the open-source smart home platform HomeAssistant, which can integrate devices from multiple manufacturers and support apps setting. As shown in Fig. 3, the smart home contains 4 testbed rooms, a toilet, a bedroom, a living room, and a kitchen. We deploy 24 IoT devices of 10 types for four rooms. Specifically, smart plugs, which are abbreviated as *P1, P2, P3, P4, P5*, are respectively used to control the water valve, a heater, TV, air conditioner, and electric kettle. There is one real resident and a part-time roommate. We deploy IoT apps following two principles: 1) popular smart apps and those basic rules provided by the IoT platform, and 2) satisfy the proposed desired automation by the resident. A total of 29 IoT apps are applied across the four testbed rooms.




| Abbr. | Device Type |
|---|---|
| MS | Motion Sensor |
| L | Light |
| THS | Temperature & Humidity Sensor |
| P | Smart Plug |
| CS | Door/Window Sensor |
| W | Window Opener |
| WS | Water Sensor |
| G | Smart Gateway |
| SA | Smoke Alarm |
| GA | Gas CO Alarm |

**Fig. 3.** Smart home with the device deployment details and device basic information.

**Data Collection and Implementation.** In our testbed, communication protocols for all IoT devices and the cloud platform include Zigbee and WiFi. To sniff all wireless network packets, we use KillBee's testing framework tool and Atmel RzRaven USB Stick to passively collect Zigbee traffic. Besides, we directly use the tcpdump tool to sniff TCP/IP packets. We collect two weeks of traffic data from the four testbed rooms. For each identified deployed device type, we manually collect device physical function-related descriptions from official device function descriptions and developer documentation of IoT platforms.

We use the NLTK tool [12] to conduct the POS tagging and lemmatization, and the networkx tool [9] for attack path discovery.

### 5.2 Results

**Threshold Selection and Device Event Fingerprints.** The burst threshold and the time threshold are important parameters in traffic analysis. First, we find an appropriate burst threshold $\lambda$, which clusters the captured packets belonging to the same device event. We respectively manually operate devices 60 times. The burst threshold is set from 0.5 to 5, with an interval of 0.25. When the burst threshold is 0.75 s, both the precision and F1 score for learning events achieve the maximum: 1 and 0.98. Table 1 illustrates the fingerprints of some device events in packet size and direction. Besides, we identify the information of the IoT devices, including the number of devices, device types, and IoT device events, from the traffic by network address and matching the device fingerprints.

The time threshold $\sigma$, which clusters device events belonging to an app behavior, impacts the effectiveness of inferring app-enabled interactions. We deployed the local IoT platform, where IoT apps and devices communicate within the LAN. We directly trigger some common apps and count the processing time and network latency between a trigger event and an action command. Most of them take less than 1s, and we set the time threshold to 1s.

**Table 1.** Examples of fingerprints of some device events.

| Device name | Communication | Events or commands | (Fingerprints) packet size with direction sequence |
|---|---|---|---|
| Aqara smart plug | Zigbee | switch.on | –48 |
| | | switch.off | –48 |
| YeelightLED Light 1S | Wifi | switch.on | –123,98,–54,82,–54 |
| | | switch.off | –124,99,–54,82,54 |
| BordLink Smart Plug | Wifi | switch.on | –130,–130,114,114 |
| | | Switch.off | –130,–130,114,114 |
| Xiaomi motion sensor | zigbee | Motion.active | 53 |
| | | motion.inactive | 53 |

**The Effectiveness of Device Interaction Identification.** Table 2 shows the specific results for each testbed room. We finally get 17 true app-enabled interactions from 29 IoT apps. The precision is 0.85 and the accuracy is 0.58. Three are false app-enabled interactions, which are caused by the living habit (e.g., the front door is typically closed right after being opened in sequential order). Besides, 12 apps are not identified for two reasons: 1) some apps are rarely triggered (e.g., "if smoke/CO is detected, open the window"), and 2) some apps are not considered in our work range (e.g., the trigger event is time).

We recognize 12 physical channels and 21 physical interactions for 21 actuators and sensors. We have manually verified that NLP tech is effective in identifying physical channels from device function descriptions. The precision is 1 and

the accuracy is 0.57. Unfortunately, the method cannot identify physical channels that are not directly related to device function (e.g., smart plugs can turn common household appliances into intelligent devices, and we can't recognize their controlled devices' physical functions).

**Table 2.** The results of device interaction identification.

| Testbed | Number of IoT apps | Number of App-enabled interactions | Number of actuators and sensors | Number of physical interactions |
|---|---|---|---|---|
| Toilet | 4 | 3 | 4 | 5 |
| Bedroom | 7 | 4 | 5 | 6 |
| Living Room | 14 | 8 | 8 | 6 |
| Kitchen | 4 | 2 | 4 | 4 |

**The Effectiveness of Attack Path Discovery.** We get a device interaction graph with 45 vertexes and 38 edges. This graph is sparse and we discover 26 potential exploitable attack paths which have more than two edges. We show some attack paths in Fig 4. We conducted two case studies to demonstrate the effectiveness of the discovered attack paths.

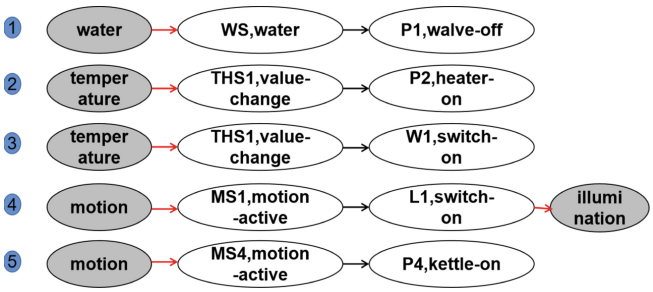



**Fig. 4.** Five potential exploitable attack paths, where the red arrows are physical interactions and black arrows are app-enabled interactions.

**Case Studies**: After we obtain paths shown in Table 4, we use the Zigbee protocol vulnerability [4] to maliciously inject events. First, we injected a THS1 value-change event with a low value, then the second path turned on the smart plug P2 which turned on the controlled heater to heat the bedroom. However, ten minutes later, the third path was triggered to open the window by a temperature-raising event. Second, when MS4 motion-active events were injected, the fourth path turned on the kettle even with no water. This could start a fire.

**Comparison with Related Work.** In Table 3, we compare our work with recent studies in multiple perspectives, e.g., scope and method. IoTMon [5] performs a static analysis of official apps and identifies hidden interactions through physical channels. ALTA [13] infers running apps from the traffic and learns context-rich information (e.g., health conditions) from apps. IoTSpy [8] analyzes the eavesdropped traffic and infers installed apps by mining IoT event dependencies. These studies provide to infer and exploit device interactions from IoT deployments. However, they do not consider physical interactions among devices and do not provide methods to discover exploitable attack paths.

**Table 3.** The comparison of D-interact with related works.

| System | IoT app inferring | Physical interactions | Attack path discovery | Real IoT deployment |
|---|---|---|---|---|
| D-interact | ✓ | ✓ | ✓ | ✓ |
| ALTA [13] | ✓ | | | ✓ |
| IoTMon [5] | | ✓ | ✓ | |
| IoTSpy [8] | ✓ | | | ✓ |
| IoTSafe [6] | | ✓ | ✓ | ✓ |

## 6 Discussion

We discuss possible countermeasures to mitigate the risk of attack path disclosure caused by traffic analysis. A straightforward solution is to eliminate statistical and temporal patterns between devices and the cloud backend by packet padding or event spoofing [11]. Then, it is hard for D-interact to identify correct events or extract correct event dependencies from encrypted traffic packets. However, they require modifications of IoT firmware and/or protocols. In addition, users should use IoT devices and home routers with high security, and update them in time to resist firmware vulnerabilities.

## 7 Conclusion

In this paper, we have designed D-interact, a new IoT attack path discovery system, which attacks realistic smart home systems by exploiting device interactions inferred from traffic and device function descriptions. We implemented D-interact on a real smart home and discovered all potential exploitable device interaction paths, which were used to conduct two real attacks. For future studies, we will focus on device interactions introduced by complex apps (e.g., the trigger is a timed event). Besides, we will design an efficient attack path search algorithm, which can quickly discover paths that can reach a specific attack target.

## References


1. Acar, A., et al.: Peek-a-boo: i see your smart home activities, even encrypted! In: Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks, pp. 207–218 (2020)
2. Home Assisant. https://www.home-assistant.io/
3. Chen, J., et al.: IoTFuzzer: discovering memory corruptions in IoT through app-based fuzzing. In: NDSS (2018)
4. Cve-poc (2021). https://github.com/chengcheng227/CVE-POC
5. Ding, W., Hu, H.: On the safety of iot device physical interaction control. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 832–846 (2018)
6. Ding, W., Hu, H., Cheng, L.: IOTSAFE: enforcing safety and security policy with real IoT physical interaction discovery. In: The 28th Network and Distributed System Security Symposium (NDSS 2021) (2021)
7. Fernandes, F., Jung, J., Prakash, A.: Security analysis of emerging smart home applications. In: 2016 IEEE Symposium on Security and Privacy (SP), pp. 636–654. IEEE (2016)
8. Gu, T., et al.: IoTSpy: uncovering human privacy leakage in IoT networks via mining wireless context. In: 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–7. IEEE (2020)
9. Hagberg, A., Swart, P., Chult, D.S.: Exploring network structure, dynamics, and function using NetworkX. Tech. rep. Los Alamos National Lab. (LANL), Los Alamos, NM (United States) (2008)
10. Lindley, D.S.: Bayesian statistics: A review. SIAM (1972)
11. Liu, X., et al.: SniffMislead: non-intrusive privacy protection against wireless packet sniffers in smart homes. In: 24th International Symposium on Research in Attacks, Intrusions and Defenses, pp. 33–47 (2021)
12. Loper, E., Bird, S.: Nltk: the natural language toolkit. arXiv preprint cs/0205028 (2002)
13. Luo, Y., et al.: Context-rich privacy leakage analysis through inferring apps in smart Home IoT. IEEE Internet of Things J. **8.4**, 2736–2750 (2020)
14. Ozmen, M.O., et al.: Discovering physical interaction vulnerabilities in IoT deployments. arXiv preprint arXiv:2102.01812 (2021)
15. Smartthing. https://www.SmartThings.com/
16. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM J. Computing **1.2**, 146–160 (1972)
17. Trimananda, R., et al.: Packet-level signatures for smart home devices. In: Network and Distributed Systems Security (NDSS) Symposium, vol. 2020 (2020)
18. Zhang, W., et al.: Homonit: monitoring smart home apps from encrypted traffic. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 1074–1088 (2018)

# A Local Rotation Transformation Model for Vehicle Re-Identification

Yanbing Chen[1(✉)], Wei Ke[1,2], Hao Sheng[3,4], and Zhang Xiong[3]

[1] Faculty of Applied Sciences, Macao Polytechnic University, Macao SAR, People's Republic of China
{yanbing.chen,wke}@ipm.edu.mo
[2] Engineering Research Centre of Applied Technology on Machine Translation and Artificial Inteligence of Ministry of Education, Macao Polytechnic University, Macao SAR, People's Republic of China
[3] State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, People's Republic of China
{shenghao,xiongz}@buaa.edu.cn
[4] Beihang Hangzhou Innovation Institute Yuhang, Yuhang District, Xixi Octagon City, Hangzhou 310023, People's Republic of China

**Abstract.** The vehicle re-identification (V-ReID) task is critical in urban surveillance and can be used for a variety of purposes. We propose a novel augmentation method to improve the V-ReID performance. Our deep learning framework mainly consists of a local rotation transformation and a target selection module. In particular, we begin by using a random selection method to locate a local region of interest in an image sample. Then, a parameter generator network is in charge of generating parameters for further image rotation transformation. Finally, a target selection module is used to retrieve the augmented image sample and update the parameter generator network. Our method is effective on VeRi-776 and VehicleID datasets, it shows that we achieve considerable competitive results with the current state-of-the-art.

**Keywords:** Vehicle re-identification · Local rotation transformation · Target selection · Parameter generator network · Local region

## 1 Introduction

Vehicle re-identification (V-ReID) [1] aims to recognize the same vehicle identity across a non-overlapping camera network. It can be used in a variety of situations [2]. For instance, V-ReID can aid law enforcement in the fight against crime. It can also assist city planners in better understanding traffic patterns [3]. Vehicle re-identification is gaining interest in the computer vision community because of its solid application background. And deep learning has become a popular and important approach [4] in the field of computer vision over the previous decade.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 76–87, 2022.
https://doi.org/10.1007/978-3-031-19208-1_7

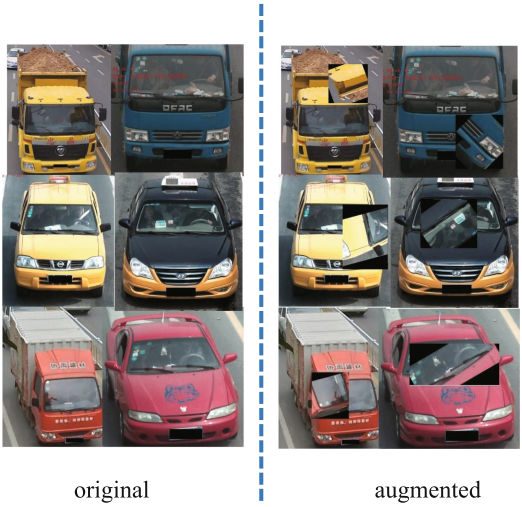



**Fig. 1.** Image augmentation results of our proposed method

The research community has been driven to create CNN-based approaches [5] for V-ReID challenges by the success of convolutional neural networks (CNN).

In order to train a deep neural network, a large amount of data is always required. Data collection and annotation, on the other hand, are prohibitively expensive. In addition, a lack of data is an obvious stumbling block when constructing a strong deep neural network. Data limitation reduces performance in the field of V-ReID as well. Data augmentation [6] is an effective way to obtain additional training samples without having to collect and annotate more data. Data warping is a common augmentation method used to generate more training samples. Geometric and color transformations, random erasing [7], adversarial training [8], and neural style transfer [9] are all examples of data warping.

Unlike previous traditional data augmentation techniques, we propose a novel data augmentation method that increases the complexity of training samples rather than the size of a dataset. Our method, in particular, combines geometric data augmentation and deep learning. In our previous work [10], we attempted to test this idea using a general framework. We adopt the architecture and reorder some of the modules to improve the efficiency. In order to increase the difficulty of network learning, we introduce a vector parameter to transform a local region of an image sample. First, we use a local region selection and a neural network known as the parameter generator network to learn the vector parameter for further data augmentation. The main goal is to keep the important information while making identification more difficult. Second, we use a target selection module in our work. This module is designed to choose the most difficult augmented images. Finally, the four augmentation images are rearranged based on the distance to the original image, then the learning parameters of parameter generate network are updated.

Extensive experiments show that our framework is extremely competitive. The augmentation results from the original images to the augmented images are shown in Fig. 1. Three datasets are subjected to a series of asymptotic ablation experiments for validation. *i.e.,* VeRi-776 [11], VehicleID [12] and VERI-Wild [13]. Our work outperforms the baseline and other previous methods, according to the experimental results. Our contributions are summarized as follows:

1. A method of data augmentation that combines geometric data augmentation with deep learning technology.
2. A deep learning framework that combines region selection with local rotation transformation.
3. Instead of adding more samples, the dataset is expanded by making the data samples more difficult. The original structure of dataset is preserved.

## 2 Related Work

### 2.1 Vehicle Re-Identification

V-ReID is intensively researched in computer vision and has a variety of important applications [14]. With the rapid development of deep learning methods in the computer vision community, neural network models are becoming a mainstream for V-ReID. V-ReID generally requires robust and discriminative image representation. Liu et al. [11] proposed the fusion of multiple features, such as colors, textures, and deep learned semantic features. Li et al. [15] introduced the deep joint discriminative learning (DJDL) model for obtaining discriminative representations from vehicle pictures. They exhibited a pipeline that used deep relative distance learning (DRDL) to map vehicle images into a euclidean space, where the distance may directly express the similarity of two vehicle images. They exhibited a pipeline that used deep relative distance learning (DRDL) to map vehicle images into a euclidean space. For solving multi-view vehicle V-ReID problems using only visual data, Zhou et al. [16] suggested a viewpoint-aware attentive multi-view inference (VAMI) model. Stevenson et al. [17], on the other hand, developed a straightforward and successful part-regularized discriminative feature-preserving technique. The method improves the ability to detect tiny variations and yields promising results.

Despite the fact that the V-ReID methods listed above differ in several respects, they all require a large number of image samples as training datasets. Unfortunately, obtaining massive datasets is difficult and expensive. As a result, data augmentation can be used to address the issue of insufficient training data for these methods.

### 2.2 Data Augmentation

Data augmentation [6] is frequently used in the training of deep neural networks to help avoid overfitting. In a specific field such as V-ReID, issues such as viewpoint, lighting, occlusion, background, and scale must be overcome. Nonetheless,

there are not many viable options for dealing with data augmentation. The goal of data augmentation is to incorporate these translational inconsistencies into the datasets so that the resulting models perform well despite these challenges. Larger datasets, it is widely assumed, result in better deep learning models. However, due to the manual labor involved in data collection and labeling, assembling large datasets can be a daunting task. Many studies on the effectiveness of data augmentation use popular academic image datasets as benchmarks.

Common augmentation methods like flipping, scaling, and perspective transformation [6] are usually useful for a single object with a static augmentation policy. The static augmentation policy, on the other hand, does not meet the dynamic optimization requirement. Cubuk et al. [18] used reinforcement learning to search for augmentation in the policy. To expedite the search process, Ho et al. [19] developed flexible augmentation policy schedules. Peng et al. [20] adversarial learning and pre-training processes to augment samples jointly.

Because our new method incorporates the previously mentioned traditional methodologies, as well as geometric and machine learning techniques, it becomes more favorable.

## 3 Methodology

We propose a local rotation transformation and a target selection module to generate more efficiently augmented image samples for the V-ReID task. In this section, the overall structure of the framework (3.1) is describe at first. Then we go through the details of the two major components individually, the Local Rotation Transformation (3.2), and the Target Selection Module (3.3).

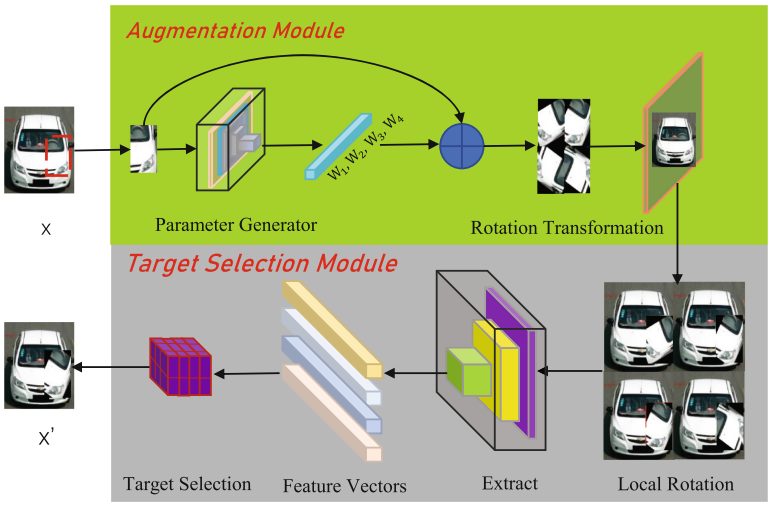



**Fig. 2.** Overview of the proposed framework. Our system has two major components: local rotation transformation (green part) and target selection module (grey part). (Color figure online)

### 3.1 Overall Framework

The suggested framework contains two components, as shown in Fig. 2, Local Rotation Transformation (LRT), and Target Selection Module(TSM).

The input of the framework is an original vehicle image sample, denoted as $x$. First, LRT takes the input image $x$ and selects one local region. Then, it produces four transformed augmented images $x_1, \ldots, x_4$, each with a different rotation parameter. Finally, given the four transformed images, the target selection module (TSM) selects the hardest sample among them, which has the largest distance from the input image $x$, as the replacement.

When the augmented images $x_1, \ldots, x_4$ pass through the selection module, it converts the augmented images into feature vectors and compare the consine distance with the original image. At the same time, the four augmented images are resorted by distance to form a new parameter vector. Note that the Parameter Generator Network (PGN) is updated by the new parameter vector which resorted by target selection module. Finally, the hardest sample selected by TSM replaces the original image as a training sample, then we go to the next iteration.

### 3.2 Local Rotation Transformation

The Local Rotation Transformation (LRT) module is designed to randomly select a local rotation region, then generates four augmented images. Three main processes are involved in this module: the Local-Region Selection, the Parameter Generator Network and the Rotation Transformation. The Local-Region Selection generates a rectangle area for rotation transformation, and the Parameter Generator Network produces the parameters for further rotation transformation. By utilizing the above rectangle area and parameters, we then generate the augmented images with the local rotation transformation process.

**Local-Region Selection.** The Local-Region Selection is utilized to randomly locate a rectangle region of interest from an image sample. The selection algorithm is shown in Algorithm 1. The area ratio of a local region is randomly initialized between $A_1$ and $A_2$, and the aspect ratio ranges from $R_l$ to $R_2$. Note that a reasonable aspect ratio can prevent selection regions from being too long or too thin. In this work, we define the area ratio as $A_t$ where $A_1 \leq A_t \leq A_2$, and the aspect ratio as $R_t$ where $R_1 \leq R_t \leq R_2$. With $A_t$ and $R_t$, we can obtain the width $W_t$ and height $H_t$ of the selected region.

**Parameter Generator Network.** After the local region is confirmed from the original image, it is fed into the Parameter Generator Network (PGN). PGN is designed to produce the four parameters for rotation transformation. The parameters, four transformation weights $W_1, \ldots, W_4$, forming the following $1 \times 4$ vector,

$$\vec{S} = \left[ W_1, W_2, W_3, W_4 \right] \tag{1}$$

**Algorithm 1.** Local-Region Selection Procedure

**Input:** image $I$;
area of image $A$;
ratio of width and height $R$;
area ratio ranges from $A_1$ to $A_2$;
aspect ratio ranges from $R_1$ to $R_2$;
**Output:** selected rectangle region

$W = I.width, H = I.height$
**while** true **do**
  $A_t = \mathbf{rand}(A_1, A_2) \times A \quad R_t = \mathbf{rand}(R_1, R_2)$
  $H_t = \sqrt{A_t \times R_t} \quad W_t = \sqrt{A_t \div R_t}$
  $P_x = \mathbf{rand}(0, W) \quad P_y = \mathbf{rand}(0, H)$
  **if** $P_x + W_t \leq W \wedge P_y + H_t \leq H$ **then**
    **return** region $(P, W_t, H_t)$
  **end if**
**end while**

**Table 1.** Architecture of the parameter generate network. **MP** denotes a $2 \times 2$ max pooling. **BN** represents the batch normalization. Height $h$ and width $w$ of the input come from the Local-Region Selection. The kernel size, stride and padding size of all the convolutional layers are respectively 3, 1 and 1.

| # | Type | Size | # | Type | Size |
|---|---|---|---|---|---|
| 1. | Input | $1 \times h \times w$ | 5. | Conv-128, ReLU, MP | $128 \times 4 \times 12$ |
| 2. | Conv-16, ReLU, MP | $16 \times 16 \times 50$ | 6. | Conv-64, BN, ReLU | $64 \times 4 \times 12$ |
| 3. | Conv-64, ReLU, MP | $64 \times 8 \times 25$ | 7. | Conv-16, BN, ReLU, MP | $16 \times 2 \times 6$ |
| 4. | Conv-128, BN, ReLU | $128 \times 8 \times 25$ | 8. | FC | 4 |

We list the architecture detail of PGN in Table 1. PGN consists of six convolutional layers and one fully connected layer. The FC layer locates at the end of the network and outputs four parameters, which represent the four weight values. We use the rotation vector and the original image to generate the augmented rotation images, see Sect. 3.2 for more details. Finally, the four augmented images are produced by LRT module. Accordingly, Parameter Generator Network is an innovative way to generate transformation parameters.

**Rotation Transformation.** After achieving the rotation vector $\vec{S}$, we adopt the Rotation Transformation to transform the original image $x$ into four augmented images $x_1, \ldots, x_4$. Rotation Transformation transforms the original image to a new augmented image, as shown in Fig. 3.

We get the rotation vector by using the four weights obtained by PGN. Then we rotate the original local image by each value in the vector. As a result, we get the local rotation image.

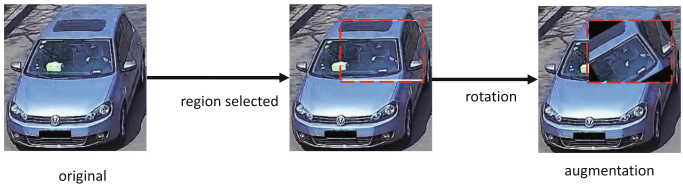



**Fig. 3.** Overview of rotation transformation

### 3.3 Target Selection Module

The Target Selection Module (TSM) is designed to select an augmented sample as hard as possible from the original sample. Meanwhile, the resort vector generated updates the PGN parameters. It is expected that an augmented image will differ from the original image as much as possible in order to improve the diversity of the training samples.

We calculate the feature distances between the augmented images $x_1, \ldots, x_4$ and the original image $x$. The most difficult augmented image with the greatest feature distance from the original image is then selected. Based on the distance, we sort the four $W_1, \ldots, W_4$, forming a new vector,

$$\vec{S^r} = \left[ W_1^r, W_2^r, W_3^r, W_4^r \right] \tag{2}$$

Note that the PGN is a neural network, the loss of PGN is described as (3), $\vec{S}$ represents the predicted value, while $\vec{S^r}$ representing the actual value. And $\alpha$ is a hyperparameter which control flexibly the loss.

$$Loss = \vec{S^r} - \vec{S} - \vec{\alpha} \tag{3}$$

Using this strategy, we choose the augmented image with the hardest one and optimize PGN. As the end, as the final augmented image, we select the hardest augmented image sample $x'$ and replace the original $x$ with $x'$ in the training dataset.

## 4 Experiment and Discussion

In this part, we construct and execute experiments on three datasets: VeRi-776, VehicleID, and VERI-Wild. Ablation experiments are designed to evaluate the performance of the three models. Also, the results are compared with those from some state-of-the-art V-ReID models.

### 4.1 Datasets

To produce comparable results, we use three well-known datasets in the V-ReID field, including VeRi-776, VehicleID and VERI-Wild. The VeRi-776 dataset is a V-ReID dataset derived from real-world surveillance scenarios. It contains over

**Table 2.** Results of Our Method on the VeRi-776, VehicleID and VERI-Wild (%)

| Model | VeRi-776 | | VehicleID | | | | | | VERI-Wild | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Small | | Medium | | Large | | Small | | Medium | | Large | |
| | R1 | mAP | R1 | mAP | R1 | mAP | R1 | mAP | R1 | mAP | R1 | mAP | R1 | mAP |
| Baseline | 95.71 | 76.59 | 83.02 | 77.02 | 80.74 | 75.04 | 79.24 | 73.98 | 93.11 | 72.60 | 90.54 | 66.51 | 86.40 | 58.52 |
| LRT (Ours) | 96.68 | 80.48 | 84.25 | 77.81 | 83.38 | 77.86 | 79.89 | 74.44 | 91.81 | 72.72 | 90.18 | 66.68 | 87.61 | 58.79 |
| LRT + TSM (Ours) | 96.88 | 81.58 | 84.31 | 77.87 | 83.46 | 77.98 | 79.92 | 74.54 | 93.37 | 72.86 | 91.06 | 66.67 | 87.75 | 58.78 |

50,000 images of 776 vehicles captured by cameras that cover one square kilometer of ground in 24 h. VehicleID is a large-scale vehicle dataset. The training set consists of 10,178 images of 13,134 vehicles, while the test set consists of 111,585 images of 13,113 other vehicles. The VERI-Wild dataset is a large-scale V-ReID dataset that was gathered in the wild. The images are captured from a large CCTV surveillance system with 174 cameras over the course of one month ($30 \times 24$ h) in unconstrained scenarios.

### 4.2 Implementation

During the data pre-processing step, all images are resized to $320 \times 320$. To test the efficacy of our data augmentation, we use the same settings as the baseline [21]. As the backbone feature network, the ResNet50 [22] is used. Soft margin triplet loss [23], which is a mini-batch of 32 images with 8 subjects and 4 images each, is used in the training of the entire network. In addition, SGD [24] is applied as the optimizer. We adopt the initial learning strategy. The learning rate begins at $10^{-2}$ and gradually decreases to $10^{-3}$ after the first ten epochs. At the $40^{\text{th}}$ epoch and the $70^{\text{th}}$ epoch, it decays to $10^{-3}$ and $10^{-4}$, respectively. In total, We train the model with 120 epochs.

The experiment adds our method as a data augmentation module into the pre-processing of the baseline [21]. See [25,26] for more details about the baseline.

### 4.3 Ablation Study

As mentioned in Sect. 3, our framework includes Local Rotation Transformation (LRT), and Target Selection Module (TSM). LRT and TSM, the relationship between them can be split, our framework can contain only LRT, and also can contain all modules. If there is only LRT, we use a random weight to do the local rotation, and then directly take this as the augmented image, without using PGN learning and TSM. Each module is monitored using a series of progressive ablation experiments.

We run experiments on three datasets with two different combinations: LRT and LRT+TSM. The results are shown in Table 2. The augmentation result is shown in Fig. 4, It is obvious that different modules produce different augmented images.

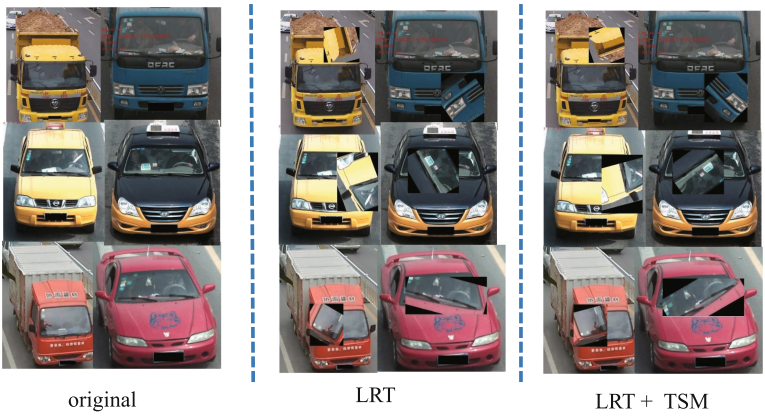



**Fig. 4.** Overview of Augmentation Transformation Combining Different Modules. **Column 1:** original images, **Column 2:** augmentation images generated by LRT, **Column 3:** augmentation images generated by LRT + TSM.

**Our Model vs. Baseline** VehicleID and VERI-Wild datasets, as we know, are further subdivided into small, medium, and large datasets.

On VeRi-776, our R1 and mAP outperform the baseline by 0.97% and 3.89%, respectively, when only LRT is used. On VehicleID, we conduct three groups of experiments. The R1 and mAP increase 1.23% and 0.79% on the SmallVehicleID group, 2.64% and 2.82% on the MediumVehicleID group, 0.65% and 0.46% on the LargeVehicleID group, respectively. Also, three groups of experiments on VERI-Wild are conducted in Table 2.

Even though the performance of those baselines are already excellent before adding TSM, our method is still higher, indicating that our method is more effective. We can observe that LRT and LRT+TSM respectively improve the mAP by 3.89% and 4.99% on VeRi-776, 0.79% and 0.85% on SmallVehicleID, 2.82% and 2.94% on MediumVehicleID, and 0.46% and 0.56% on LargeVehicleID. We find that the mAP has been improved the most apparently on MediumVehicleID.

Also, as shown in Table 2, LRT and LRT+TSM improve the mAP respectively on VERIWild. All of our methods significantly improves the performance over the baseline. As we can see, LRT is the main component of our methods. When used in conjunction with TAM, performance ramps up.

### 4.4 Comparison with the State-of-the-Art

On VeRi-776 and VehicleID, the performance of our method is compared against state-of-the-art. as shown in Table 3 and Table 4 respectively. The results show that our method outperforms other methods significantly. In comparison to all other methods, our scheme outperforms them on the three datasets VeRi-776 and VehicleID. Our method outperforms the second best method in mAP accuracy on VeRi-776 by 2.18%, and it also performs well on VehicleID.

**Table 3.** Comparison with the Sate-of-the-Art V-ReID methods on VeRi (%)

| Methods | mAP | R1 | R5 |
|---|---|---|---|
| BS [27] | 67.55 | 90.23 | 96.42 |
| CCA [28] | 68.05 | 91.71 | 94.34 |
| PRN [29] | 70.2 | 92.2 | 97.9 |
| AGNET [30] | 71.59 | 95.61 | 96.56 |
| VehicleX [31] | 73.26 | 94.99 | 97.97 |
| MDL [21] | 79.4 | 90.7 | – |
| Ours | **81.58** | **96.88** | **98.94** |

**Table 4.** Comparison of the State-of-the-Art V-ReID Methods on VehicleID (%)

| Methods | Small | | Medium | | Large | |
|---|---|---|---|---|---|---|
| | R1 | R5 | R1 | R5 | R1 | R5 |
| AGNET [30] | 71.15 | 83.78 | 69.23 | 81.41 | 65.74 | 78.28 |
| AAVER [32] | 74.7 | 93.8 | 68.6 | 90.0 | 63.5 | 85.6 |
| CCA [28] | 75.51 | 91.14 | 73.60 | 86.46 | 70.08 | 83.20 |
| PRN [29] | 78.4 | 92.3 | 75.0 | 88.3 | 74.2 | 86.4 |
| BS [27] | 78.80 | 96.17 | 73.41 | 92.57 | 69.33 | 89.45 |
| VehicleX [31] | 79.81 | 93.17 | 76.74 | 90.34 | 73.88 | 88.18 |
| Ours | **84.31** | **93.43** | **83.46** | **90.58** | **79.92** | **86.47** |

## 5 Conclusion

In this work, we present a novel method for augmenting image data in V-ReID tasks. To ensure that as much noise information as possible is added to the image sample, a local rotation transformation and a target selection module are used. Rather than increasing the number of training samples, we increase the difficulty of samples. Actually, our method can be used as a pre-processing layer in other deep learning systems, broadening its application potential. Unlike previous frameworks, we target local regions of images and use convolutional operations to transform them in order to increase the difficulty of the network and thus improve its performance.

**Acknowledgments.** This study is partially supported by the National Key R&D Program of China (No.2018YFB2100800), the National Natural Science Foundation of China (No.61 872025), and Macao Polytechnic University (Research Project RP/ESCA-03/2020), and the Open Fund of the State Key Laboratory of Software Development Environment(No. SKLSDE-2021ZX-03) Thanks for the support from HAWKEYE Group.

## References


1. Khan, S.D., Ullah, H.: A survey of advances in vision-based vehicle re-identification. Comput. Vis. Image Underst. **182**, 50–63 (2019)
2. Xu, H., Cai, Z., Li, R., Li, W.: Efficient citycam-to-edge cooperative learning for vehicle counting in its. IEEE Trans. Intell. Trans. Syst. **23**, 16600–16611 (2022)
3. Xiong, Z., Cai, Z., Han, Q., Alrawais, A., Li, W.: Adgan: protect your location privacy in camera data of auto-driving vehicles. IEEE Trans. Indust. Inf. **17**, 6200–6210 (2020)
4. Wang, J., Cai, Z., Yu, J.: Achieving personalized $k$-anonymity-based content privacy for autonomous vehicles in cps. IEEE Trans. Industr. Inf. **16**(6), 4242–4251 (2020)
5. Xiong, Z., Xu, H., Li, W., Cai, Z.: Multi-source adversarial sample attack on autonomous vehicles. IEEE Trans. Veh. Technol. **70**(3), 2822–2835 (2021)
6. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. J. Big Data **6**(1), 60 (2019)
7. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34(7) (2017)
8. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses, arXiv preprint arXiv:1705.07204 (2017)
9. Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., Song, M.: Neural style transfer: a review. IEEE Trans. Visual Comput. Graphics **26**(11), 3365–3385 (2019)
10. Chen, Y., et al.: Local perspective based synthesis for vehicle re-identification: a transformation state adversarial method. J. Vis. Commun. Image Rep. **83**, 103432 (2022)
11. Liu, X., Liu, W., Ma, H., Fu, H.: Large-scale vehicle re-identification in urban surveillance videos. In: 2016 IEEE International Conference on Multimedia and Expo (ICME), vol. 22, pp. 1–6. IEEE (2016)
12. Liu, H., Tian, Y., Yang, Y., Pang, L., Huang, T.: Deep relative distance learning: tell the difference between similar vehicles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2167–2175 (2016)
13. Lou, Y., Bai, Y., Liu, J., Wang, S., Duan, L.: Veri-wild: A large dataset and a new method for vehicle re-identification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3235–3243 (2019)
14. Cai, Z., Zheng, X., Yu, J.: A differential-private framework for urban traffic flows estimation via taxi companies. IEEE Trans. Ind. Inf. **15**, 6492–6499 (2019)
15. Li, Y., Li, Y., Yan, H., Liu, J.: Deep joint discriminative learning for vehicle re-identification and retrieval. In: 2017 IEEE International Conference on Image Processing (ICIP), vol. 2017, pp. 395–399. IEEE (2017)
16. Zhou, Y., Liu, L., Shao, L.: Vehicle re-identification by deep hidden multi-view inference. IEEE Trans. Image Process. **27**(7), 3275–3287 (2018)
17. Stevenson, J.A., Sun, X., Mitchell, N.C.: Despeckling srtm and other topographic data with a denoising algorithm. Geomorphology **114**(3), 238–252 (2010)
18. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: learning augmentation strategies from data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 113–123 (2019)
19. Ho, D., Liang, E., Chen, X., Stoica, I., Abbeel, P.: Population based augmentation: Efficient learning of augmentation policy schedules. In: International Conference on Machine Learning, pp. 2731–2741. PMLR (2019)

20. Peng, X., Tang, Z., Yang, F., Feris, R.S., Metaxas, D.: Jointly optimize data augmentation and network training: adversarial data augmentation in human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2226–2234 (2018)
21. He, S., et al.: Multi-domain learning and identity mining for vehicle re-identification. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 582–583 (2020)
22. Jung, H., Choi, M.-K., Jung, J., Lee, J.-H., Kwon, S., Young Jung, W.: Resnet-based vehicle classification and localization in traffic surveillance systems. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 61–67 (2017)
23. Yuan, Y., Chen, W., Yang, Y., Wang, Z.: In defense of the triplet loss again: Learning robust person re-identification with fast approximated triplet loss and label distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 354–355 (2020)
24. Zhang, S., Choromanska, A., LeCun, Y.: Deep learning with elastic averaging sgd, arXiv preprint arXiv:1412.6651 (2014)
25. Luo, H., Gu, Y., Liao, X., Lai, S., Jiang, W.: Bag of tricks and a strong baseline for deep person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2019)
26. Luo, H., et al.: A strong baseline and batch normalization neck for deep person re-identification. IEEE Trans. Multimedia **22**(10), 2597–2609 (2019)
27. Kuma, R., Weill, E., Aghdasi, F., Sriram, P.: Vehicle re-identification: an efficient baseline using triplet embedding. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–9. IEEE (2019)
28. Peng, J., Jiang, G., Chen, D., Zhao, T., Wang, H., Fu, X.: Eliminating cross-camera bias for vehicle re-identification. Multimedia Tools Appli. **81**, 1–17 (2020)
29. He, B., Li, J., Zhao, Y., Tian, Y.: Part-regularized near-duplicate vehicle re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3997–4005 (2019)
30. Zheng, A., Lin, X., Li, C., He, R., Tang, J.: Attributes guided feature learning for vehicle re-identification, arXiv preprint arXiv:1905.08997 (2019)
31. Yao, Y., Zheng, L., Yang, X., Naphade, M., Gedeon, T.: Simulating content consistent vehicle datasets with attribute descent, arXiv preprint arXiv:1912.08855 (2019)
32. Khorramshahi, P., Kumar, A., Peri, N., Rambhatla, S.S., Chen, J.-C., Chellappa, R.: A dual-path model with adaptive attention for vehicle re-identification. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 6132–6141 (2019)

# A Prototype System for Blockchain Performance Evaluation

Kaixiang Hou[1], Tianyi Xu[1], Chao Xu[1], Xiaobo Zhou[1], Tie Qiu[1(✉)], and Fengbiao Zan[2]

[1] College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
{hkx_98,tianyi.xu,xuchao,xiaobo.zhou,qiutie}@tju.edu.cn
[2] The Computer College of Qinghai Minzu University, Qinghai 810007, China

**Abstract.** Due to complex blockchain programs and numerous blockchain nodes, it takes a huge amount of time and economic cost to conduct blockchain experiments. Existing open source projects do not support modifications to the underlying blockchain, and existing blockchain simulators only focus on a single blockchain system and cannot flexibly extend or replace models. Regarding the issues above, this paper proposes a prototype system for blockchain performance evaluation, including real deployment test and simulation test. In real deployment test, a five-layer architecture for building a lightweight and efficient testing system is proposed. And in simulation test, a general scheme for building blockchain simulator is proposed, which can realize the test of throughput, storage allocation and reputation management. Experiments show that the prototype system proposed in this paper can effectively improve the efficiency of blockchain performance evaluation.

**Keywords:** Blockchain · Prototype system · Simulation test · Real deployment test

## 1 Introduction

Bitcoin, which is based on blockchain technology, has led a wave of cryptocurrency [1]. With the introduction of smart contracts, the application fields of blockchain have been further expanded [2], such as evidence-based traceability [3], data sharing [4], industrial Internet of Things [5], privacy computing [6] and other fields. Blockchain ensures the safe operation of Decentralized Applications (DApps), but with the explosive growth of DApps, more and more transactions are pouring into the blockchain network. Since transactions need to be stored and reach a consensus in the entire network in order, the blockchain technology has encountered bottlenecks in throughput, transaction confirmation speed, and data storage.

In order to optimize the performance of blockchain, scholars have made significant achievements in consensus algorithms, blockchain structures, cryptology, etc. However, when these achievements are put into practice, they encountered


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 88–100, 2022.
https://doi.org/10.1007/978-3-031-19208-1_8

a key problem, that is, the evaluation of blockchain performance. Newly developed blockchain projects should be deployed into testing environment to evaluate performance before being deployed into production environment [7]. However, deploying blockchain projects for evaluation is difficult. If we deploy only a few nodes in Ethereum, it is difficult to acquire information about the entire network. If a dedicated network for evaluation is constructed, although the information of the entire network can be obtained, it costs too much.

Since open source projects such as Ethereum, Hyperledger Fabric are all designed for business, not designed to evaluate the performance of blockchain, it is difficult to carry out underlying experiments of blockchain, such as modifying the architecture of Peer to Peer (P2P) network and the consensus algorithms. Therefore, a blockchain real deployment testing system that is lightweight, has comprehensive interfaces for test, and can be quickly deployed on large-scale nodes is urgently needed.

The scale of nodes in DApps is usually huge, especially in public chain applications, and the blockchain program needs to be frequently modified and redeployed on the nodes during the research and development, which increases the cost of purchasing and managing servers. In order to evaluate blockchain proformance faster with less cost, while guaranteeing the accuracy of experimental results, some scholars have begun to study blockchain simulators, trying to reproduce the operating status of a real blockchain system by designing and running simulation models. For example, SimBlock [8], VIBES [9] and Bitcoin mining Simulater [10] have made great attempts in designing blockchain simulators. However, the simulation technology of blockchain is still in the initial stage, and the existing simulators have very limited functionality [11]. Most simulators can only simulate the variant algorithm of Proof of Work (PoW) or Practical Byzantine Fault Tolerance (PBFT) but not customized algorithms. This is also the reason why existing simulators cannot provide substantial assistance to the research and development of blockchain. Therefore, we need a general scheme of carrying out simulation test.

This paper proposes a prototype system for blockchain performance evaluation, including real deployment test and simulation test. The main contributions of this paper are as follows:

1. The scheme and prototype system of real deployment test are proposed. Users can flexibly configure the parameters of the consensus algorithm through the system, and quickly deploy algorithms to all blockchain nodes. Besides, throughput, the status of blockchain nodes, and the detailed information of block can be monitored in real time.
2. The scheme and prototype system of simulation test are proposed which can realize the test of throughput, storage allocation and reputation management on large-scale nodes. Users can configure experimental parameters through the system, such as the number of nodes and committee nodes, network bandwidth to carry out simulation tests and monitor the experimental results.

The remainder of this paper is organized as follows. Section 2 introduces the related work on blockchain performance evaluation. Section 3 explains the

scheme of building prototype system. Section 4 illustrates the architecture of prototype system based on the above schemes and discusses the experimental results. Section 5 elaborates the summary.

## 2 Related Work

In this section, we analyze the related works on blockchain performance evaluation from two parts: the real deployment test and the simulation test.

There are many open source blockchain projects, such as Ethereum, EOS, Hyperledger Fabric, FISCO and Xuperchain. These projects all support smart contracts, and DApps can be easily developed by them, but they only have a few consensus algorithms to choose from. It is difficult to modify the underlying consensus algorithm on them. And the above projects are designed for application development and mainly focus on the construction of virtual machine and ensuring the high availability of the system, resulting in a huge client size. But for blockchain performance evaluation, the most important is to improve the testing efficiency of the consensus algorithm. Moreover, it is necessary to implement a OAM (Operation And Maintenance) system to realize the rapid deployment and update of the consensus algorithm on large-scale nodes, thereby improving the efficiency of experiment. However, the above systems do not have the OAM system for blockchain nodes.

For the real deployment test, this paper propose a lightweight construction scheme and a prototype system of blockchain, which only retains the necessary part of conducting test and adds the OAM system to manange nodes.

Shadow-Bitcoin [12] and VIBES [9] try to simulate the broadcast of blocks in the bitcoin network. Gervais proposes a quantitative framework to analyze the impact of various consensus algorithms and network parameters on system security and performance [13]. Aoki proposes a blockchain network simulator called SimBlock, which can easily change the behavior of nodes, so that the impact of node behavior on the blockchain can be studied [8]. Faria proposes Blocksim to rapidly model the blockchain and has studied the impact of doubling the number of transactions and encrypted communication on block propagation delays [14]. Dinh proposes the evaluation framework blockbench to analyze the performance of private blockchains [15]. Wuthier implements the visualization of the PoW [16]. However, these works only focus on a single blockchain system and cannot flexibly extend or replace the blockchain models.

For the simulation test, this paper also proposes a general scheme and a prototype system of building blockchain simulator.

## 3 The Scheme of Building Prototype System

In this section, we give our schemes of building prototype system for blockchain performance evaluation.

### 3.1 The Scheme of Real Deployment Test

The real deployment test requires us to deploy the blockchain program into the physical nodes or virtual machines. However, building a complete blockchain system is time-consuming and will seriously affect the research progress. Moreover, we need to continuously adjust the blockchain program and redeploy it into the heterogeneous blockchain nodes, which brings challenges to node management. Therefore, an efficient real deployment testing system is required to be lightweight and manageable. In view of the above requirements, we give the following scheme, as shown in Fig. 1.

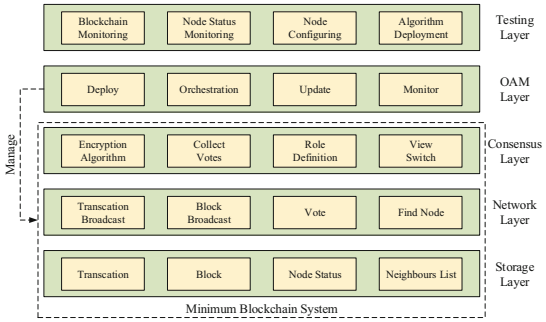



**Fig. 1.** Architecture of real deployment test.

The scheme of real deployment test is divided into five layers, from the bottom up, the storage layer, the network layer, the consensus layer, the OAM layer and the application layer, covering the minimum blockchain system and the management system of blockchain nodes.

The storage layer is responsible for caching transactions, storing blocks and node status such as online time, reputation value and the list of neighbors. Non-relational Database (NOSQL) and catch database can be used to store data, such as using redis to cache transactions and recently packaged blocks, because when verifying new blocks, most relevant transactions will be found from recently packaged blocks for validation. It can reduce the read and write burden of database if recently packaged blocks are stored in cache. Mongodb, a NOSQL, can be used to store block data, because the amount of block data is large, and block structures will be dynamically adjusted due to the switching of consensus algorithms. If relational database is used to store blocks, this will result in frequent modification of table structure, which makes managing the database become cumbersome. However, Mongodb does not need to change table structure because it uses collection to stroge data.

The network layer is responsible for discovering neighbors, driving the network to initiate and forward transactions and blocks, and collecting votes to reach consensus. Dstributed Hash Table (DHT) or Kademlia [17] can be employed to discover neightbors. We may also need to fix the topology of the

network to compare the impact of other factors on blockchain performance by manually assigning neighbors to nodes. The protocol for data interaction can use the Transmission Control Protocol (TCP) like Bitcoin and Ethereum.

The consensus layer includes four elements to realize a basic consensus algorithm: node role division, encryption algorithm, view interaction and view switching. Node role division is used to elect consensus nodes and leader nodes. Encryption algorithms are used to encrypt signatures. Opinion interaction refers to the exchange of opinions between consensus nodes to reach consensus. And view switching refers to the re-election of the master node when consensus fails due to the downtime of the master node. The above three layers are the minimum components of a blockchain system. We can build a lightweight blockchain system without smart contract virtual machine according to the above ideas.

The OAM layer is designed to facilitate the management of blockchain nodes and improve the efficiency of experiments, which plays an important role in blockchain test, especially in large-scale test. The core idea of OAM layer is to containerize the blockchain program, and perform container deployment, orchestration, update, and monitoring. By packaging the blockchain program into a container and starting it, we can deploy blockchain program without complex environment configuration, such as configuring the database and cache. The OAM layer also includes a mirror, which is responsible for communicating with other mirrors, obtaining the running status and passing the data to the testing layer for processing.

The testing layer is used to call the OAM system and the blockchain system for performing experiments. Users can easily configure the parameters of the consensus algorithm, and deploy the new consensus algorithms through the OAM layer into each blockchain node. At the same time, it can monitor the performance of the blockchain such as throughput, transaction confirmation speed, the status of the node, the change of node reputation, and the storage overhead.

### 3.2 The Scheme of Real Simulation Test

This paper proposes the following general scheme to map the blockchain in the real world to the simulation environment, as shown in Fig. 2. Nodes and blocks will be mapped as objects in the simulation environment with methods for packing, broadcasting, validating and storing blocks, etc. After the object executes method, it will cause the status of the objects to change. For example, after the node calls the method of packing blocks, the usage of CPU will increase. By observing and analyzing the change of status, the performance of blockchain such as throughput can be calculated.

To further illustrate how we build the simulation model, Algorithm 1 about calculating throughput in PoW consensus is given and the variables used are explained in Table 1.

To calculate throughput, we simulate the process of block packaging, broadcasting and validation. Before the simulation algorithm runs to the termination time, the status of each node in the system will be updated every $s$ time. The

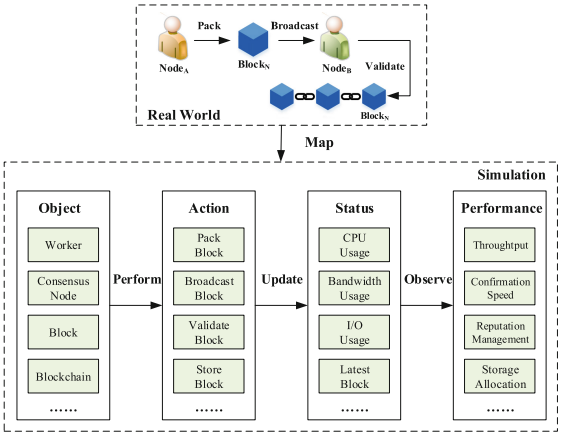



**Fig. 2.** A general scheme of simulation test.

shorter the $s$, the more accurate the simulation results are, but the longer the time spent in simulation will be.

First, the algorithm will calculate and record $H_{N_i}$ up to the current moment. If $H_{N_i}$ exceeds $H_d$, $N_i$ is considered to have successfully min a block and $H_{N_i}$ will be reset. After that, $L_{N_i}$ will be increased by one and new *btasks* will be added to $Task^b_{N_i}$ to simulate the process of broadcasting block. Once a *btask* is completed, new *btasks* will be added to the $Task^b$ of $N_{btask_r}$, in order to simulate the process of forwarding the received block. It is also necessary to add

**Table 1.** Table of variables

| Variables | Description |
|---|---|
| $s$ | The status of blockchain is updated every $s$ time |
| $d$ | The difficult of mining block |
| $N_i, i \in [0, n)$ | Blockchain node |
| $t^c_{N_i}$ | Time cost for $N_i$ to perform a hash operation |
| $T_e$ | The end time of the simulation |
| $T_p$ | The elapsed time of the simulation |
| $H_{N_i}$ | The number of times $N_i$ has performed the hash operations |
| $H_d$ | The number of hash operations needed to mine a block at $d$ |
| $L_{N_i}$ | The height of the latest block of $N_i$ |
| $Task^b_{N_i}$ | $N_i$'s task list of broadcasting block |
| $Task^v_{N_i}$ | $N_i$'s task list of validating block |
| $btask$ | A task of broadcasting block |
| $vtask$ | A task of validating block |
| $btask_r$ | The target node of $btask$ |
| $vtask_h$ | The height of the block that validated by $vtask$ |
| $M$ | The average number of transactions contained in one block |

**Algorithm 1.** Calculating throughput in PoW consensus

**Input:** $T_e$,$d$
**Output:** $Throughput$

**while** $T_p < T_e$ **do**
  **for** $i = 0; i < n; i++$ **do**
    $H_{N_i} += \frac{s}{t^c_{N_i}}$
    **if** $H_{N_i} > H_d$ **then**
      $H_{N_i} = 0$, $L_{N_i} += 1$
      add $btasks$ to $Task^b_{N_i}$
    **end if**
    **for** each $btask \in Task^b_{N_i}$ **do**
      **if** $btask$ is finished **then**
        add new $block$ received from the $btask$ to $Task^v_{N_{btask_r}}$
        add $btasks$ to $Task^b_{N_{btask_r}}$
        delete $btask$ from $Task^b_{N_i}$
      **end if**
    **end for**
    **for** each $vtask \in Task^v_{N_i}$ **do**
      **if** $vtask$ is finished and validated **then**
        $L_{N_i} = max(L_{N_i}, vtask_h)$
        delete $vtask$ from $Task^v_{N_i}$
      **end if**
    **end for**
    $process(Task^b_{N_i}, Task^v_{N_i})$
    $T_p = T_p + s$
  **end for**
**end while**
add $L_{N_i}$ to $L$ for all $i \in [0, n)$
$L.sort()$
$Throughput = \frac{L[0.2 \cdot n] \cdot M}{T_e}$
**return** $Throughput$

the received block to the $Task^v$ of $N_{btask_r}$ to simulate the process of validating block. Once a $vtask$ is completed, $L_{N_i}$ will be updated based on the $vtask_h$. Next, we need to process the $Task^b_{N_i}$ and $Task^v_{N_i}$ according to the CPU and bandwidth usage of $N_i$.

After simulation, the algorithm will record $L_{N_i}$ of each node in a list and sort the list in ascending order. We consider the blocks that have been synchronized by more than 80% nodes to be valid blocks, and the throughput can be calculated according to the number of valid blocks. The running time of the algorithm is mainly affected by $T_e$, $s$ and $n$. The time complexity of the algorithm is $\Theta(\frac{T_e}{s} \cdot n)$.

It should be noted that the time consumed by some processes, such as validating blocks, generating signatures and storing blocks, are difficult to calculate. However, we can obtain the time consumed by these processes through real deployment tests and map the time to the number of steps in the simulation space.

# 4 Experiment

## 4.1 The Architecture of Prototype System

Based on the scheme given in Sect. 3, we develop a prototype system for blockchain performance evaluation, as shown in Fig. 3.

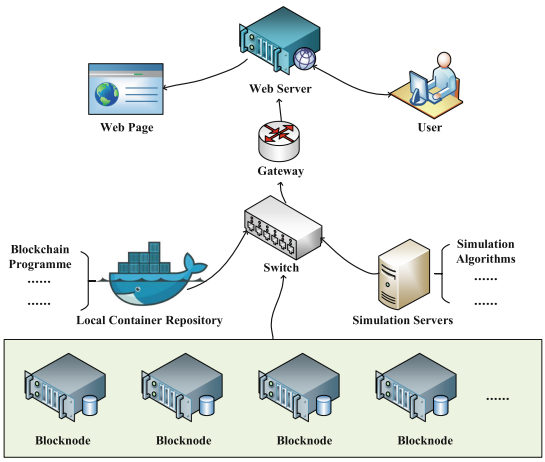



**Fig. 3.** The architecture of prototype system.

The blockchain nodes run the blockchain program in the form of a docker container, and pulls the latest blockchain program image from the local image repository according to the user's command. The local container repository maintains historical versions of blockchain images, and users can push the modified blockchain programs to the local container repository at any time. The files of simulation algorithms and the calling interfaces are deployed in the simulation testing server. Blockchain nodes, container repository, and simulation servers are connected to the same switch and exchange data with web server through gataway. We can modify the configuration of consensus algorithms, conduct experiments and observe the system status in real time through the web pages.

## 4.2 The Experiments of Real Deployment Test

**Blockchain Status Monitoring.** It is used to monitor the block height, number of nodes, total amount of data and specific information of the block in real time. At the same time, the CPU, memory and disk usage of web server can also be monitored, as shown in Fig. 4.

**Blockchain Performance Test.** It is used to configure the parameters of consensus algorithms and monitor the changes of the blockchain throughput and node status such as CPU usage and memory usage in real time, as shown in Fig. 5.

In real deploymeng test, the mining difficulty is defined as $\alpha$, and the text of block header is defined as $s_b$. $SHA256(s)$ represents computing the 256-bit hash string of $s$. $Cal0(s)$ represents counting the number of leading zeros in $s$. The process of mining can be described by (1), that is, if $s_b$ satisfies (1), the block is packaged successfully. We can observe a significant increase in throughput as $\alpha$ is reduced from 4 to 2. And as the throughput increases, CPU and memory usage also increases.

$$Cal0(SHA256(SHA256(s_b))) > \alpha \tag{1}$$

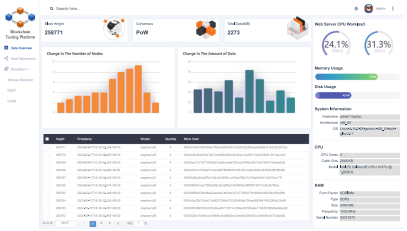


**Fig. 4.** Blockchain data monitoring.


**Fig. 5.** Blockchain performance test.

### 4.3 The Experiments of Simulation Test

**Storage Allocation Test.** This algorithm is suitable for scenarios which multiple nodes jointly store a complete piece of data, and used to allocate storage comprehensively considering data availability, storage speed and storage cost. We designed storage allocation algorithms [18] based on Genetic Algorithm (GA) and NSGA2 [19]. Users can modify the relevant parameters of the algorithms such as population size, crossover probability, mutation probability and iteration times from the web pages, and observe the optimization process of the algorithm. The results of storage allocation are shown in Fig. 6.

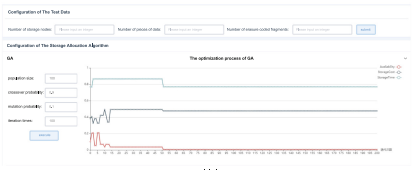


**Fig. 6.** Storage allocation test.

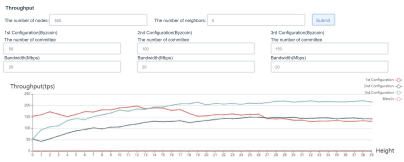


**Fig. 7.** Throughput test.

**Throughput Test.** We have implemented the simulation of the PoW consensus algorithm and the Byzcoin [20], and users can modify the parameters of the consensus algorithms such as the number of nodes in the entire network, committee nodes and neighbor nodes and network bandwidth on the web pages while observing the changes of throughput, as shown in Fig. 7. We also compare the results of simulation test with the real deployment test in PoW consensus. The throughput of the two tests is very close as shown in Fig. 8. To calculate the accuracy of simulation test, we define the results of simulation test as $X = \{x_1, x_2, \ldots, x_m\}$, the results of real deployment test as $Z = \{z_1, z_2, \ldots, z_m\}$, and the deviation of simulation results as $\Theta$. $\Theta$ can be calculated by (2). After calculation, the deviation of the simulation test is relatively small, close to 3.47%.

$$\Theta = \frac{\sqrt{\frac{\sum_{i=1}^{m}(x_i - \bar{z})^2}{m-1}}}{\bar{z}} \times 100\% \tag{2}$$

**Node Reputation Management Test.** A blockchain storage allocation algorithm based on node reputation is implemented [21]. Users can configure the performance parameters of each node on the web pages, such as hard disk capacity, CPU frequency, read and write rate and bandwidth which have an effect the basic reputation of each node. At the same time, the interaction process between nodes including initiating transactions, packaging blocks and storing blocks will be simulated which influences the growth reputation of each node. Growth reputation and underlying reputation are summed according to the weight to get the reputation of each node, and the storage allocation is carried out according to the reputation value. We can observe the changes of the reputation and the result of storage allocation on the web pages. The experimental results are shown in Fig. 9.

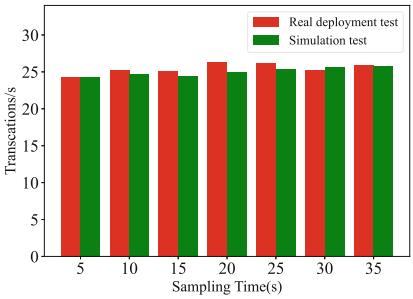



**Fig. 8.** Comparison of throughput.

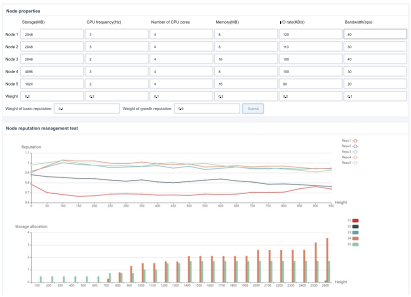


**Fig. 9.** Reputation management test.

## 5 Conclusion

This paper proposes a prototype system for blockchain performance evaluation, including real deployment test and simulation test. In real deployment test, the consensus algorithm can be flexibly configured through the web pages and automatically deployed on blockchain nodes. Users can monitor blockchain status in real time. In simulation test, the throughput test, storage allocation test, and reputation management test are implemented. Users can configure experimental parameters through the system, such as node size and network bandwidth to conduct simulation tests and monitor the experimental results. Experiments show that the scheme proposed in this paper can effectively reduce the time required to conduct blockchain experiments.

**Acknowledgements.** This work is supported by the National Key R&D Program of China (No.2019YFB1703601), the Joint Funds of the National Natural Science Foundation of China (No.U2001204), Key R&D Program of Tianjin (No.20YFZCGX01150), the Tianjin Science Foundation for Distinguished Young Scholars (No.20JCJQJC00250) and the Open Research of Zhejiang Lab (No.2021KF0AB02).

## References

1. Hou, B., Chen, F.: A study on nine years of bitcoin transactions: understanding real-world behaviors of bitcoin miners and users. In: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), pp. 1031–1043. IEEE (2020)
2. Wang, S., Ouyang, L., Yuan, Y., Ni, X., Han, X., Wang, F.Y.: Blockchain-enabled smart contracts: architecture, applications, and future trends. IEEE Trans. Syst. Man Cybern. Syst. **49**(11), 2266–2277 (2019)
3. Xu, H., Zhang, L., Onireti, O., Fang, Y., Buchanan, W.J., Imran, M.A.: Beeptrace: blockchain-enabled privacy-preserving contact tracing for covid-19 pandemic and beyond. IEEE Internet Things J. **8**(5), 3915–3929 (2020)
4. Chi, J., et al.: A secure and efficient data sharing scheme based on blockchain in industrial internet of things. J. Netw. Comput. Appl. **167**, 102710 (2020)

5. Shen, M., et al.: Blockchain-assisted secure device authentication for cross-domain industrial IoT. IEEE J. Sel. Areas Commun. **38**(5), 942–954 (2020)
6. Weng, J., Weng, J., Zhang, J., Li, M., Zhang, Y., Luo, W.: Deepchain: auditable and privacy-preserving deep learning with blockchain-based incentive. IEEE Trans. Dependable Secure Comput. **18**(5), 2438–2455 (2019)
7. Foytik, P., Shetty, S., Gochhayat, S.P., Herath, E., Tosh, D., Njilla, L.: A blockchain simulator for evaluating consensus algorithms in diverse networking environments. In: 2020 Spring Simulation Conference (SpringSim), pp. 1–12. IEEE (2020)
8. Aoki, Y., Otsuki, K., Kaneko, T., Banno, R., Shudo, K.: Simblock: A blockchain network simulator. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 325–329. IEEE (2019)
9. Stoykov, L., Zhang, K., Jacobsen, H.A.: Vibes: fast blockchain simulations for large-scale peer-to-peer networks. In: Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Posters and Demos, pp. 19–20 (2017)
10. Carlsten, M., Kalodner, H., Weinberg, S.M., Narayanan, A.: On the instability of bitcoin without the block reward. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 154–167 (2016)
11. Paulavičius, R., Grigaitis, S., Filatovas, E.: An overview and current status of blockchain simulators. In: 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 1–3. IEEE (2021)
12. Miller, A., Jansen, R.: {Shadow-Bitcoin}: Scalable simulation via direct execution of {Multi-Threaded} applications. In: 8th Workshop on Cyber Security Experimentation and Test (CSET 2015) (2015)
13. Gervais, A., Karame, G.O., Wüst, K., Glykantzis, V., Ritzdorf, H., Capkun, S.: On the security and performance of proof of work blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 3–16 (2016)
14. Faria, C., Correia, M.: Blocksim: blockchain simulator. In: 2019 IEEE International Conference on Blockchain (Blockchain), pp. 439–446. IEEE (2019)
15. Dinh, T.T.A., Wang, J., Chen, G., Liu, R., Ooi, B.C., Tan, K.L.: Blockbench: a framework for analyzing private blockchains. In: Proceedings of the 2017 ACM International Conference on Management of Data, pp. 1085–1100 (2017)
16. Wuthier, S., Chang, S.Y.: Proof-of-work network simulator for blockchain and cryptocurrency research. In: 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), pp. 1098–1101. IEEE (2021)
17. Maymounkov, P., Mazières, D.: Kademlia: a peer-to-peer information system based on the XOR metric. In: Druschel, P., Kaashoek, F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 53–65. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45748-8_5
18. Xu, T., Qiu, T., Hu, D., Mu, C., Wan, Z., Liu, W.: A scalable two-layer blockchain system for distributed multi-cloud storage in iiot. IEEE Trans. Industrial Inf. (2022)

19. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Schoenauer, M., et al. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45356-3_83
20. Kogias, E.K., Jovanovic, P., Gailly, N., Khoffi, I., Gasser, L., Ford, B.: Enhancing bitcoin security and performance with strong consistency via collective signing. In: 25th Usenix Security Symposium (Usenix Security 2016), pp. 279–296 (2016)
21. Fan, Y., et al.: Dlbn: group storage mechanism based on double layer blockchain network. IEEE Internet of Things J. (2022)

# A Spatial-Temporal Convolutional Model with Improved Graph Representation

Yang Lv[1], Zesheng Cheng[1(✉)], Zhiqiang Lv[1,2], and Jianbo Li[1]

[1] College of Computer Science and Technology, Qingdao University, Qingdao 266071, China
czs_110@hotmail.com
[2] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

**Abstract.** Traffic prediction problem plays a crucial role in the research of intelligent transportation systems. Traffic flow is an important indicator to measure the traffic status. Traffic flow prediction can not only provide a scientific basis for traffic managers but also support other road services. This work proposes a spatial-temporal convolutional neural network with improved graph representation (IGR-TCN) for predicting urban traffic flow, which solves the limitations of traditional methods considering only a single road section or a single detector. IGR-TCN reduces the computational complexity by using a convolutional structure, the temporal convolution layer uses dilated convolution, and causal convolution to optimize the long-term prediction capability. The graph representation proposed in this work improves the existing spatial-temporal correlation model and increases the spatial correlation trend of the data. The IGR-TCN fits better than traditional recurrent neural networks, traditional graph convolution models, and graph spatial-temporal models. It can be more effective for spatial-temporal information prediction.

**Keywords:** Intelligent transportation systems · Traffic prediction · Graph convolution · Temporal Convolution

## 1 Introduction

The Intelligent Transportation Systems (ITS) [1] is an effective way to reduce traffic pressure and solve the problems of frequent traffic accidents and environmental pollution, which is important for economic development. Therefore, spatial-temporal sequence prediction for graph data has gradually become the focus and difficulty of ITS research. Unstructured data [2] do not have the same spatial localization. It not only has an arbitrary data range but also a complex topology that does not satisfy translation invariance. It is difficult to define convolution kernels in non-Euclidean data. Nevertheless, unstructured data is a valuable asset that has long been underutilized and contains a large amount of valuable information. In dealing with the complex spatial-temporal relationships that exist in unstructured data, non-Euclidean data can be extracted into a graph structure. Graphs usually contain relations between vertices and edges, and also contain each vertex's features. The purpose of deep learning on graphs is to learn these two features. The


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 101–112, 2022.
https://doi.org/10.1007/978-3-031-19208-1_9

introduction of graph neural networks [3, 4] allow for better modeling of graph structure and contextual information in traffic systems. Nowadays, there are deep learning methods [5–12] that integrate spatial-temporal properties and learn the spatial-temporal coupling features of sequences by introducing spatial matrices. However, they generally consider a single attribute. Only one of the two, connectivity or weight, is considered to build the graph structure for calculation. The problem of how to combine the graph connectivity and weight information to better capture the potential relationships of data in the traffic network is still to be solved.

Traffic data shows obvious spatial and temporal correlation characteristics, and the current traffic data of road sections will be influenced by other road sections nearby, in addition to the dependency relationship with historical data, the scientific estimation of traffic data development trend can be realized by establishing its temporal and spatial relationship model [13, 14]. The use of scientific methods to predict real-time traffic is important for improving people's comfort and relieving urban traffic congestion. Spatial-temporal sequences are difficult to be described by linear processes, and the applicability of traditional methods is limited when dealing with data having obvious nonlinear relationships. With the popularity of deep learning, spatial-temporal sequence models based on deep learning have gained wide attention to further explore the spatial-temporal relationships in urban traffic and provide new solutions for traffic prediction.

Based on the problems of unstructured data in traffic networks and computing time-dependent sequence features mentioned above, a new graphical spatial-temporal network IGR-TCN is proposed in this work. The specific contributions are as follows.

(1) In this work, we adopt a new method for unstructured data processing to obtain the best representation in space and thus preserve the local affinity of the graph by considering the local relationships and global relative importance of the nodes. The method considers both the connectivity and weight information of the graph, which can perceive the potential relationships of the data more intuitively in space.
(2) In this work, we design a spatial-temporal model for traffic flow data prediction, which overcomes the problems of RNN [15] not supporting data-parallel processing and slow training speed. The multilayer residual structure is used instead of the selected-pass structure, the network has a flexible perceptual field, and the causal convolution can track more distant historical information for long-term traffic prediction.
(3) In real datasets, IGR-TCN is compared with traditional recurrent neural networks, graph convolution (GCN) [4], and graphical spatial-temporal networks, and the experimental results show that the IGR-TCN model has performance advantages of around 5%.

## 2 Related Work

### 2.1 Data Presented with Graph

From the perspective of spatial-temporal sequence data, the spatial-temporal sequence prediction methods based on deep learning of traffic can be divided into two methods: spatial-temporal prediction for grid data and spatial-temporal prediction for graph data.

The grid spatial-temporal prediction algorithm is only applicable to deal with spatial-temporal data in Euclidean space. For most of the unstructured data in real life, we use graph data-oriented spatial-temporal prediction algorithms. In the graph data-oriented spatial-temporal sequence prediction problem, we extract the traffic data as a graph structure for input and compute it by adding an adjacency matrix, which is considered the key to capturing spatial correlation in traffic prediction. The adjacency matrix of the temporal graph convolutional network (TGCN) [5] consists of 0s and 1s, and it performs well in long-term prediction, but the edge values and local peaks appear to be underfitting the prediction results. Both the spatial-temporal graph convolutional network (STGCN) [6] and the attention-based spatial-temporal graph convolutional network (ASTGCN) [7] construct adjacency matrices based on distance, and the adjacency matrices constructed in this way are static and do not take into account the similarity of traffic states between long-distance nodes. Li et al. proposed the diffusion convolutional recurrent neural network (DCRNN) [9], which uses directed graphs to represent pairs of traffic sensors between spatial correlations and an encoder-decoder mechanism to compute temporal correlations, the nodes of this graph are sensors, and the edge weights represent the proximity between sensors measured by the road network distance, and the DCRNN has a better improvement compared to the experimental baseline. The graph convolutional neural network with a data-driven graph filter (GCNN-DDGF) [10] proposes three patterns including distance, interaction, and correlation graphs to build adjacency matrices, and then the three matrices are fused into a new matrix to perform subsequent convolution operations.

### 2.2 Spatial-Temporal Data Forecasting

For the traffic flow prediction problem of graphs, we use the historical traffic state on the graph to make predictions. For long-term traffic prediction, STGCN operates on traffic data time series based on graph neural network and replaces RNN structure completely with convolutional structure, which is more accurate and faster to train prime than traditional statistical and machine learning methods, but the extraction of temporal features is cruder. ASTGCN models three temporal features of traffic flow separately, and its temporal attention mechanism can effectively capture the dynamic spatial-temporal correlation of traffic data and has a better performance compared to STGCN on real highway traffic datasets. The hybrid spatial-temporal graph convolution network (H-STGCN) [11] proposed by Dai et al. converts upcoming traffic volumes into equivalent proceeding times using graph convolution to capture the spatial dependence and the composite adjacency matrix to capture the intrinsic features of traffic approximation, H-STGCN takes advantage of the significant advantages offered by future data to consistently outperform STGCN. The TGCN model combines a GCN and a gating unit (GRU) for traffic prediction, where the GCN computes spatial correlation while the GRU obtains time-dependent information by computing the time-series variation of the input signal.

# 3 Data Design

## 3.1 Unstructured Graph Data

A new graph representation is used in the IGR-TCN model, considering both graph weights and connectivity information, using the distance matrix as the edge weights of the graph and the degree matrix as the point weights of the graph, considering the connectivity structure of the graph, and keeping the localization of the graph in space. In the construction of the graph adjacency matrix, both the local relationships between nodes and the global relative importance of nodes are considered. The weights of the nodes are used to measure the local nature of the nodes, and the connectivity is used to measure the importance of the nodes in the overall structure. The optimized objective equation is as follows.

$$L = \Phi - \frac{\Phi P + P^T \Phi}{2} \tag{1}$$

$P$ is the weight matrix, which is the distance between two locations, and $\Phi$ is the connectivity matrix, which is the degree diagonal matrix of the vertices. The global impact of connectivity on the relationship through this optimization objective reflects the local nature reflected by the edges between nodes. The generalized eigenvector composition graph structure is solved by generalized Eigen decomposition as follows.

$$Ly = \lambda \Phi y \tag{2}$$

# 4 Model Design

The overall structure of the IGR-TCN model is shown in Fig. 1. Firstly, the adjacency matrix $A$ of the traffic road network is established according to the distance and connectivity between cities, and then the traffic flow data are processed and involved in the first layer of time convolution $TC^1$, whose main role is to scale compression and feature transformation of traffic flow data. The adjacency matrix $A$ is combined with the first layer of time convolution to calculate the spatial correlation of traffic flow. The second temporal convolution layer calculates the temporal correlation of traffic flow, and the temporal convolution consists of multilayer residual blocks (containing causal and dilated convolution). The IGR-TCN model also uses BatchNorma2D to prevent gradient disappearance and gradient explosion of the convolutional network, and finally, the fully connected layer maps the data features to the sample space to obtain the predicted traffic flow and calculates the model metrics.

## 4.1 Spatial Convolutional Layer

Spatial convolution calculates the spatial correlation of urban traffic flows and builds the graph structure based on the distance and connectivity between locations. In graph convolution, each node of the graph is affected by neighboring nodes and further nodes.

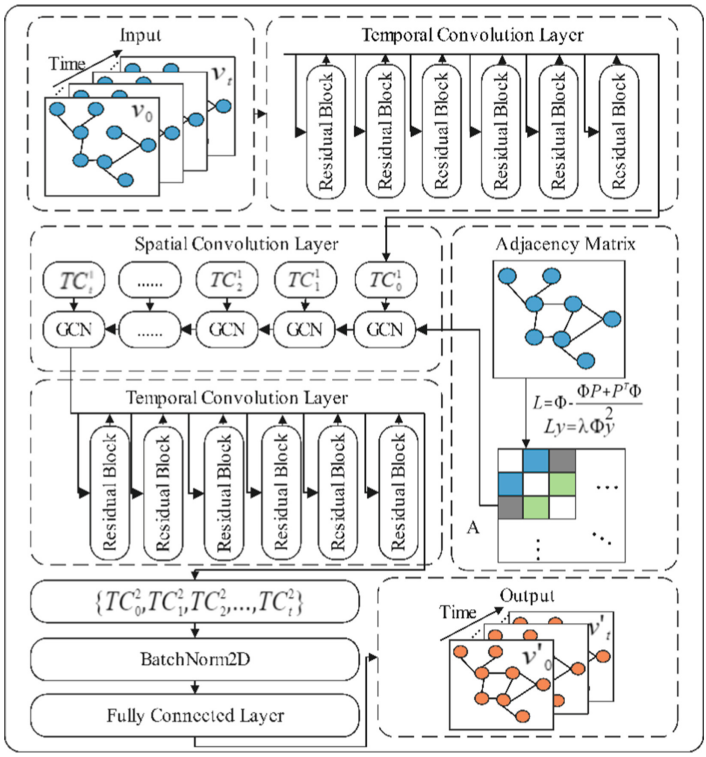



**Fig. 1.** Structure of the IGR-TCN.

The new graph representation used maintains the localization of the vertices in the graph to all their neighbors, considering both local properties and global relationships. The Fourier transform makes possible convolution on the graph, whereas in the time domain it is not possible to determine a convolution kernel to perform the convolution operation on the graph due to its arbitrarily complex topology. If the transform is applied to the frequency domain, the convolution operation is easily performed. The signal on the graph is generally expressed as follows.

$$x = [x_1, x_2, \ldots, x_n]^T \in R^n \tag{3}$$

The $n$ is the number of nodes with a signal value at each node and the value of the i-th node is $x(i) = x_i$. The graph Fourier transform uses our new combinatorial matrix with its eigenvectors as the basis functions of the graph Fourier transform. The signal on an arbitrary graph can be expressed as follows.

$$x = \hat{x}(\lambda_1)\overrightarrow{u_1} + \hat{x}(\lambda_2)\overrightarrow{u_2} + \cdots + \hat{x}(\lambda_n)\overrightarrow{u_n} \tag{4}$$

The $U = (\overrightarrow{u_1}, \overrightarrow{u_2}, \ldots, \overrightarrow{u_n})$ is the eigenvector of the combined matrix, then the graph Fourier transform is shown as follows.

$$\hat{x} = U^T x \tag{5}$$

To improve the efficiency of the decomposition of the graph structure matrix in large graphs, the computational procedure of the K-order Chebyshev approximate adjacency matrix is used.

$$g(\theta) \cdot y \approx \sum_{i=1}^{K} \theta_i \cdot T_i(A) \cdot y \tag{6}$$

The $y$ is the output of the first layer of temporal convolution and $A$ is the graph structure matrix. The calculation using the K-local convolution algorithm effectively reduces the number of parameters and ensures that only values in the $K$ range are considered for the current node.

### 4.2 Temporal Convolutional Layer

To improve the accuracy of predicting long-term traffic flow, we design a new temporal convolution layer to compute time-dependent features. Whose main effects are mainly reflected in retaining historical information while using causal convolution to compute long-term historical information; using dilation convolution to increase the perceptual field of convolution, and improving the problem of the small perceptual field due to changes in the convex function of the expansion factor, which can capture multi-scale contextual. The use of dilation convolution increases the perceptual field problem due to the variation of the convex function of the expansion factor and can capture multi-scale contextual information without causing local information loss in deeper computation. It does not need to process the data sequentially like RNN and uses a multi-layer residual structure instead of a selected-pass structure to improve the training speed.

Based on the two principles that temporal convolutional networks produce the same length as the output and that future information will not be leaked, the temporal convolutional layers are implemented using 1D complete convolution and causal convolution, respectively. Causal convolution is used to track longer historical information, and the output at time $t$ is only related to the previous layer at time $t$ and earlier convolution, i.e., if the input sequence at the current layer is $[0, i]$, then the input sequence at the next layer will become $[0, i + 1]$.

Simple causal convolution can only recall historical information that presents linearity in the depth of the network. To handle those sequence tasks that require longer histories, dilation convolution is used to achieve an exponential increase in the perceptual field. Dilated convolution guarantees the expansion of the receptive field with the loss of resolution. The dilation convolution equation is as follows.

$$y_t = \sum_{i=1}^{K} f_i \cdot x_{t-i \cdot d} \tag{7}$$

The $d$ is the expansion coefficient, and $t - i \cdot d$ explains the past direction. Setting different $d$ will make the receptive field different, and the expansion convolution is the regular convolution when $d = 1$. The range of the perceptual field can be increased by increasing $d$ and $K$. The expansion factor varies according to the convex function, and we can change the expansion factor according to the convex function to solve the problem that the information obtained by long-distance convolution is not related to the deepening of the network layer, which ensures that the local information will not be lost. This approach limits the range of the depth network perceptual field to a certain extent, making the model more accurate in the process of depth feature calculation.

# 5 Experiments

## 5.1 Dataset

We validated our model on two California highway datasets, PeMSD4 and PeMSD8 [7]. The datasets are based on raw data obtained by sampling at a frequency of 30 s/time aggregated into samples with 5-min intervals and contain characteristics of three dimensions: traffic flow, average speed, average lane occupancy, and geographic location information of the detectors that collect this information.

## 5.2 Settings

This experiment validates the traffic flow data for 300 locations of PeMSD4 and 97 given locations of PeMSD8, respectively, dividing the training and testing sets, with 80% of the data used for model training and 20% of the data used for testing. The original traffic flow data were scaled to the range $[-1, 1]$ using min-max normalization, and the output values were rescaled to the original range of values in the evaluation phase. Considering the training cost of the model, the convolution kernel size $K$ of the graph convolution layer is set to 3. For all experiments, five historical data are used in this work to predict the future traffic flow after 15, 30, and 45 min.

This experiment uses two metrics to evaluate the performance of the model, including the mean square error (RMSE) and the mean absolute error (MAE).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2} \tag{8}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{y_i} - y_i\right| \tag{9}$$

The RMSE is the average of the squared differences between the predicted and true values, followed by the square root, which measures the average size of the errors and the deviation between the observed and true values. The RMSE is very sensitive to values that are far from the overall data and can be a good indicator of the accuracy of the prediction results.

Besides, we set up three baselines to validate the model on the same dataset, which includes the traditional recurrent neural network, graphical convolution, and graphical spatial-temporal neural network. The structure of each model is as follows.

LSTM: long short-term memory network, a recurrent neural network for processing sequential data, controls the transmission state of data through a selective pass mechanism.
GCN: Graph Convolutional Network, a neural network architecture for processing graph data, uses a filter with shared parameters to constitute a new feature representation by weighted summation of pixel points in spatial regions.
STGCN: Spatial-temporal graph convolutional network, which does not use regular convolution and recursive units, consists of stacked spatial-temporal convolutional blocks and output layers.

ASTGCN: Attention-based spatial-temporal graph convolutional network, consisting of three independent blocks of the same structure modeling each of the three temporal characteristics of traffic flow (recent, daily, and weekly cycles).

### 5.3 Quantitative Analysis

The results of the two metrics of each model for different datasets are shown in Table 1 and Table 2, and the evaluation metrics are poor for datasets with more locations. The worst performance of LSTM indicates that LSTM cannot handle complex spatial data. The GCN evaluation metrics are improved compared to LSTM, which indicates that considering only spatial characteristics can improve the prediction accuracy to some extent. STGCN and ASTGCN are graphical spatial-temporal neural networks, compared with LSTM and GCN, which use GCN to calculate spatial correlation and recurrent neural networks to calculate temporal correlation, and they have the disadvantage that they cannot access deep multi-scale contextual features of the data. For all models, the average prediction error of the model increases with the prediction time, and the IGR-TCN model is the model with the lowest prediction error growth rate, which means that the IGR-TCN model is more robust than benchmark models.

IGR-TCN not only considers the spatial information of the data but also the temporal correlation of the data. The error reduction rate compared with ASTGCN is shown that IGR-TCN can better capture the spatial-temporal characteristics and can well handle the long-term traffic flow prediction problem with higher prediction accuracy.

**Table 1.** Results of evaluation metrics on PeMSD4 dataset.

| Model | RMSE | | | MAE | | |
|---|---|---|---|---|---|---|
| | 15 min | 30 min | 45 min | 15 min | 30 min | 45 min |
| LSTM | 78.61 | 89.11 | 98.28 | 59.50 | 68.46 | 73.38 |
| GCN | 62.91 | 64.53 | 66.46 | 47.06 | 48.44 | 49.69 |
| STGCN | 52.57 | 54.92 | 56.85 | 38.93 | 40.16 | 41.64 |
| ASTGCN | 48.98 | 50.51 | 52.73 | 35.72 | 37.57 | 39.02 |
| **IGR-TCN** | **47.78** | **48.08** | **48.97** | **34.72** | **34.94** | **35.66** |

### 5.4 Qualitative Analysis

In this work, further qualitative analysis of the data from the PeMSD8 dataset is performed. The LSTM prediction values are less effective than IGR-TCN as can be seen more intuitively in the scatter plot in Fig. 2(b). LSTM is a modified recurrent neural network with a gradient disappearance problem that prevents the network from learning long-term dependencies, and although forgetting gates, input gates, and output gates are added to reduce this problem, LSTM is still not a perfect solution. From Fig. 2(c) and

**Table 2.** Results of evaluation metrics on PeMSD8 dataset.

| Model | RMSE | | | MAE | | |
|---|---|---|---|---|---|---|
| | 15 min | 30 min | 45 min | 15 min | 30 min | 45 min |
| LSTM | 61.71 | 65.69 | 69.67 | 49.48 | 50.18 | 52.10 |
| GCN | 50.81 | 52.42 | 54.12 | 35.61 | 36.82 | 38.15 |
| STGCN | 44.80 | 45.20 | 46.37 | 31.85 | 32.24 | 33.44 |
| ASTGCN | 43.75 | 44.42 | 45.91 | 31.56 | 31.79 | 32.95 |
| **IGR-TCN** | **41.15** | **41.92** | **41.97** | **29.81** | **30.38** | **30.63** |

2(d), it can be seen that the IGR-TCN model is more suitable for the predicted values of the traffic peaks than GCN, which is because the GCN does not consider the temporal correlation of the data, resulting in poor prediction results. In Fig. 2(e), 2(f), 2(g), and 2(h), although the prediction effect of STGCN and ASTGCN is not as good as IGR-TCN, they have introduced graph convolution to suppress the edge under-fitting phenomenon and significantly improve prediction for larger and smaller traffic values. In addition, these two models fit the predicted values better when there are large changes in the real data, which is due to the introduction of RNN to solve the local peak under-fitting problem brought by GCN.

We also do two kinds of ablation experiments of IGR-TCN and compare them with the IGR-TCN model, which removes different convolution layers, IGR-TCN-NS indicates that the effect of the spatial convolution layer is not considered, and IGR-TCN-NT indicates that the effect of temporal convolution layer is not considered. Our improved graphical spatial-temporal model IGR-TCN with a temporal convolution layer establishes a wide range of perceptual fields through dilated convolution, overcomes the long-term temporal dependence on historical data through a multilayer residual structure, and better captures the temporal correlation of traffic data, which can be verified by the fitting results in Fig. 3(a), where the IGR-TCN-NS prediction results are poor, suggesting that the addition of temporal convolution layer plays an important role in model prediction. The IGR-TCN-NT model in Fig. 3(c) fits better than the IGR-TCN-NS model in Fig. 3(a), and the predicted values in Fig. 3(d) are closer to the true values than those in Fig. 3(b), indicating that the spatial convolution layer has a more obvious effect on the model performance improvement than the temporal convolution layer. From the comparison between Fig. 3(c) and Fig. 2(c), we know that the IGR-TCN-NT model obtains relatively perfect curve fitting results, which confirms that the spatial information of graphs can be better preserved by incorporating graph connectivity information into the data. The predicted values of IGR-TCN-NT in Fig. 3(d) are closer to the diagonal than those of GCN in Fig. 2(d), indicating that our improved graph representation method considers both weight and graph connectivity information and has better prediction results. The experimental results show that the IGR-TCN can effectively predict spatial-temporal information.

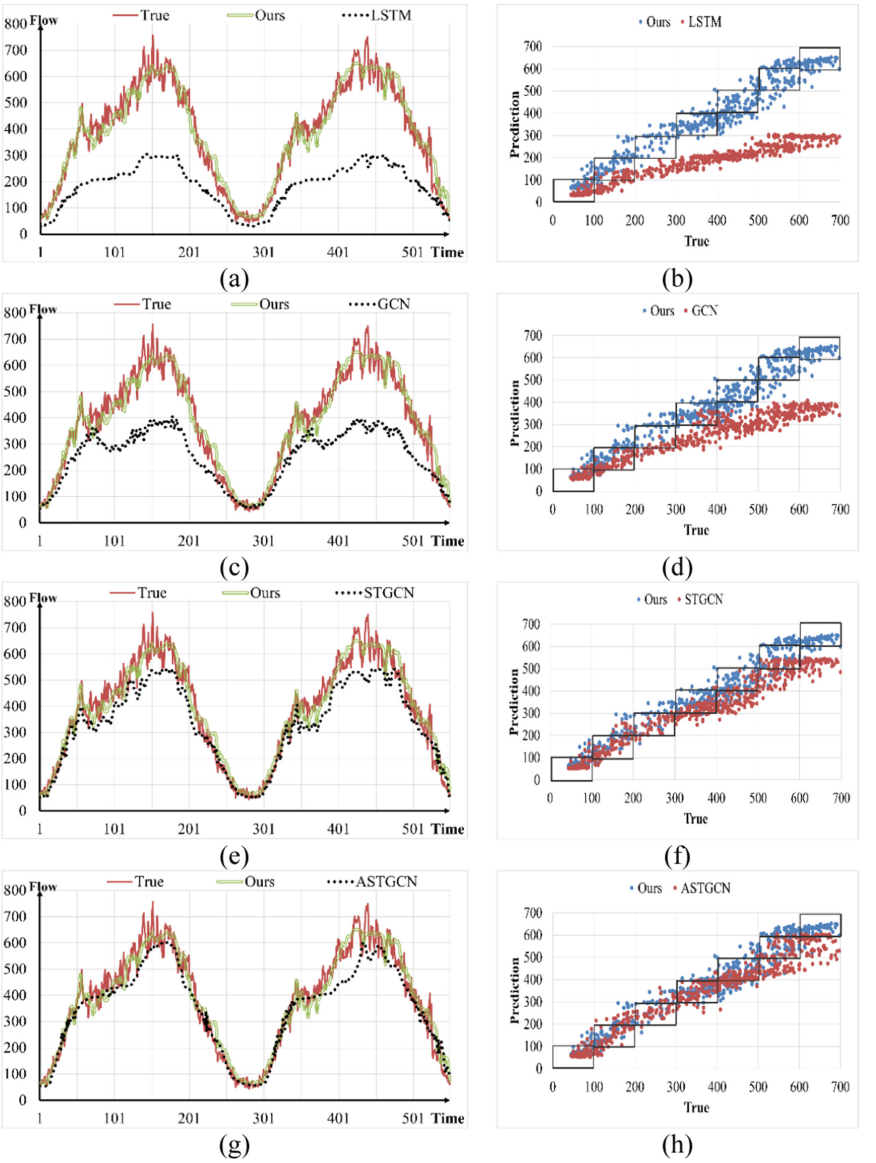



**Fig. 2.** The prediction effect of each model is shown. (a) (c) (e) (g) shows the variation of individual nodes with time, the horizontal coordinates indicate the time and the vertical coordinates indicate the traffic flow values. (b) (d) (f) (h) are scattered plots of the prediction of each model, the horizontal coordinates indicate the real values and the vertical coordinates indicate the predicted traffic values.

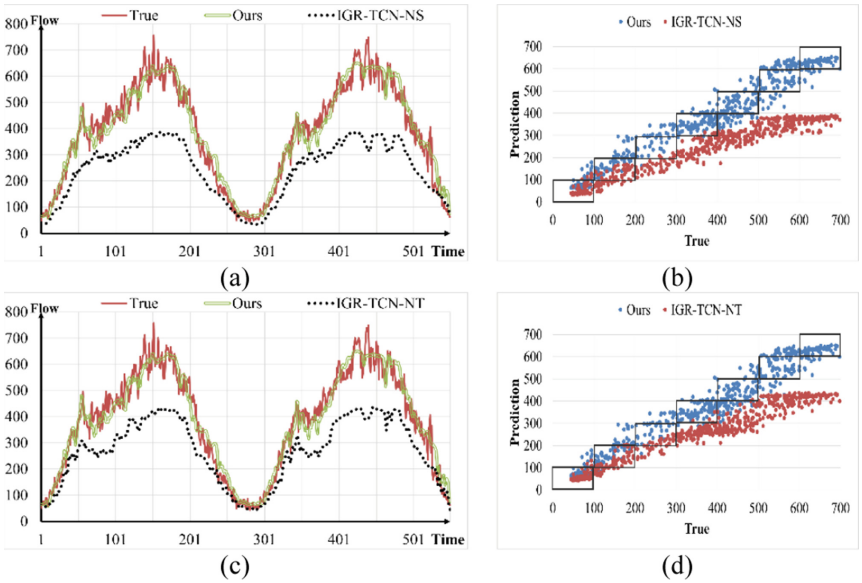



**Fig. 3.** Results of IGR-TCN ablation experiments. (a) (b) shows the variation of individual nodes with time, the horizontal coordinates indicate the time and the vertical coordinates indicate the traffic flow values. (c) (d) shows the predicted scatter plot for each model, the horizontal coordinate indicates the true value and the vertical coordinate indicates the predicted traffic value.

## 6 Conclusions

Nowadays, serious traffic condition including frequent congestion and accidents becomes a problem for all mankind. It was known to all long ago that road construction blindly is not an efficient method to solve it. Therefore, the development of ITS in the city is necessary. In this work, a new deep learning spatial-temporal model IGR-TCN is proposed to predict future urban traffic flows using traffic data from real traffic networks, relying on graph connectivity and weights to build graph structures that can more fully perceive the potential relationships of data in the spatial convolution layer, while the temporal convolution layer makes full use of the temporal correlation of traffic flows. Comparison with LSTM, GCN, STGCN, and ASTGCN benchmark models reveals that IGR-TCN has higher prediction accuracy, which verifies that the model has advantages in capturing spatial-temporal features and temporal correlations. In future work, we will further improve the traffic map representation to further enhance its role in the modeling process under the condition of considering multiple factors.

## References

1. Sumalee, A., Ho, H.W.: Smarter and more connected: future intelligent transportation system. IATSS Res. **42**(2), 67–71 (2018)
2. Zhang, C., Song, D., Huang, C., et al.: Heterogeneous graph neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp.793–803. Association for Computing Machinery, Online (2020)

3. Chiang, W.L., Liu, X., et al.: Cluster-GCN: an efficient algorithm for training deep and large graph convolutional networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, pp. 257–266. Association for Computing Machinery (2019)
4. Lv, Z., Li, J., Li, H., et al.: Blind travel prediction based on obstacle avoidance in indoor scene. In: Wireless Communications and Mobile Computing 2021, (2021)
5. Chen, B., Guo, W., Tang, R., et al.: TGCN: tag graph convolutional network for tag-aware recommendation. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp.155–164. Association for Computing Machinery, Online (2020)
6. Tang, J., Liang, J., Liu, F., et al.: Multi-community passenger demand prediction at region level based on spatio-temporal graph convolutional network. Transp. Res. Part C Emerg. Technol. **124**, 102915 (2021)
7. Guo, S., Lin, Y., Feng, N., et al.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, Hawaii, pp. 922–929. AAAI Press (2019)
8. Lv, Z., Li, J., Dong, C., et al.: Deep learning in the COVID-19 epidemic: a deep model for urban traffic revitalization index. Data Knowl. Eng. **135**, 101912 (2021)
9. Li, Y., Yu, R., Shahabi, C., et al.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
10. Lin, L., He, Z., Peeta, S.: Predicting station-level hourly demand in a large-scale bike-sharing network: graph convolutional neural network approach. Transp. Res. Part C Emerg. Technol. **97**, 258–276 (2018)
11. Dai, R., Xu, S., Gu, Q., et al.: Hybrid spatio-temporal graph convolutional network: improving traffic prediction with navigation data. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 3074–3082. Association for Computing Machinery, Online (2020)
12. Lv, Z., Li, J., Dong, C., et al.: DeepSTF: a deep spatial–temporal forecast model of taxi flow. Comput. J. (2021)
13. Gu, Z., Saberi, M., Sarvi, M., et al.: A big data approach for clustering and calibration of link fundamental diagrams for large-scale network simulation applications. Transp. Res. Procedia **23**, 901–921 (2017)
14. Zhang, X., Liu, W., Waller, S.T., et al.: Modelling and managing the integrated morning-evening commuting and parking patterns under the fully autonomous vehicle environment. Transp. Res. Part B Methodol. **128**, 380–407 (2019)
15. Xu, Z., Lv, Z., Li, J., et al.: A novel perspective on travel demand prediction considering natural environmental and socioeconomic factors. IEEE Intell. Transp. Syst. Mag. 2–25 (2022)

# A Proof-of-Weighted-Planned-Behavior Consensus for Efficient and Reliable Cyber-Physical Systems

Fang Ouyang[1], Zheng Bao[1], Lixiao Zhou[2], Feilong Lin[1], Zhaolong Hu[1], Changbing Tang[2(✉)], and Minglu Li[1]

[1] School of Mathematics and Computer Science, Zhejiang Normal University, Jinhua, China
{ouyangfang,zsdbaozheng,bruce_lin,huzhaolong,mlli}@zjnu.edu.cn
[2] School of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua, China
{zhou_lixiao,tangcb}@zjnu.edu.cn

**Abstract.** Recently, blockchain was introduced into the cyber-physical systems, which provides services of privacy and trust. However, reliability and system performance issues exist when blockchain and cyber-physical systems are integrated. In this paper, we design a blockchain-enabled cyber-physical system, where a new blockchain consensus is used to solve the problems of reliability and system performance. Firstly, an autonomous consensus mechanism called Proof-of-Weighted-Planned-Behavior is established based on the theory of planned behavior. Then, the behavior of consensus participants gets further explained by introducing credit evaluation and vulnerable node analysis. Moreover, considering the Jain fairness index, a dynamic authorizer group mechanism that coordinates reliability and decentralization is developed. By optimizing the credit threshold of the authorization group, the security and reliability of our designed mechanism are guaranteed. Finally, the experimental simulation results prove that compared with the traditional consensus, our proposed consensus improves the reliability and the system performance of the blockchain-enabled cyber-physical systems.

**Keywords:** Blockchain · Cyber-physical systems · Proof-of-Weighted-Planned-Behavior · Consensus mechanism

## 1 Introduction

Cyber-physical systems (CPS) are essentially a distributed control system [1], which consists of collaborative computing objects associated with the physical world and its processes [2]. Recently, as adistributed ledger database, blockchian [3] has been introduced into various scenarios, such as smart grid [4], energy trading [5], smart industry [6], etc. In particular, the blockchain was introduced into the research of CPS. The application fields of blockchain CPS include: smart


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 113–125, 2022.
https://doi.org/10.1007/978-3-031-19208-1_10

grid [7], smart city [8], energy systems [6], etc. Blockchain and CPS integration can increase its level of reliability without any trusted third party [8]. The most valuable advantage of blockchain is to provide a platform of trust, where all transaction records will remain unchanged in a decentralized manner [9].

Take microgrid energy trading [10] as an example, the integration of blockchain with CPS has two major issues: On one hand, numerous malicious transactions will occur, which damages the reliability of the system. On the other hand, the degree of dispersion will not be guaranteed [11], and the performance of the system will generally be low [12].

For these two issues, some researchers have proposed ways to overcome them. Yang T et al. [13] based on blockchain technology, proposed a new high-confidence data trusted collection model for trust issues. Moreover, Proof-of-trust (PoT) [11] uses the credit value of nodes as the standard for social evaluation. However, both of these will lead to centralization problems [14]. Laszka A et al. [14] provided an innovative algorithms, which solved three challenges i.e.trust, privacy and centralization [2]. However, they all suffer from low system performance issues [13]. A promising protocol, practical byzantine fault tolerance (PBFT), provides a fast energy transaction process and supports scalability using Hyperledger [15]. However, reliability can also be an issue if the validator is poorly chosen. In [5], where consensus decisions are performed by a small group of carefully selected validators. Meanwhile, trust is an issue because producers or consumers can break committed energy transfers [4]. Proof-of-planned-behavior (PoPB) [16] consensus can solve the problems of reliability and decentralization in microgrid energy trading, but PoPB does not take into account the weight factor. In fact, the weight factor is an important quantity in theory of planned behavior(TPB), which can affect the behaviors of nodes in energy trading process.

In order to solve the above challenges [14], we continue our work in [16] and design a new blockchain consensus mechanism based on TPB with weight, called Proof-of-Weighted-Planned-Behaviour (PoWPB), for microgrid energy trading in CPS. By selecting high reputation as the authorization group candidate, our proposed PoWPB consensus can greatly increase the reliability of energy trading, and improve the system performance. The contributions of this work are presented as follows:

- We design a new consensus based on TPB with weight, which is called PoWPB consensus. The new consensus can improve the reliability and system performance of the integration of blockchain and CPS.
- We analyze the vulnerability of the system to increase the reliability and stability of the authorization group when a node become an authorization candidate node. Moreover, we determine the number of authorized candidate nodes to achieve the best system performance through an optimization problem.
- Considering the node trading choice preference, we consider the weight ratio when a node applies to become a member of a high-reputation authorization group. Besides, we refer to the Jain fairness index to ensure the system achieve an optimal balance between trustworthiness and dispersion.

## 2 Architecture of Blockchain-Enabled CPS

The integration of blockchain and CPS is shown in Fig. 1, we specify an example of microgrid energy trading. The system is divided into three layers: 1) A real space layer, 2) A cyber space layer, 3) A blockchain layer.

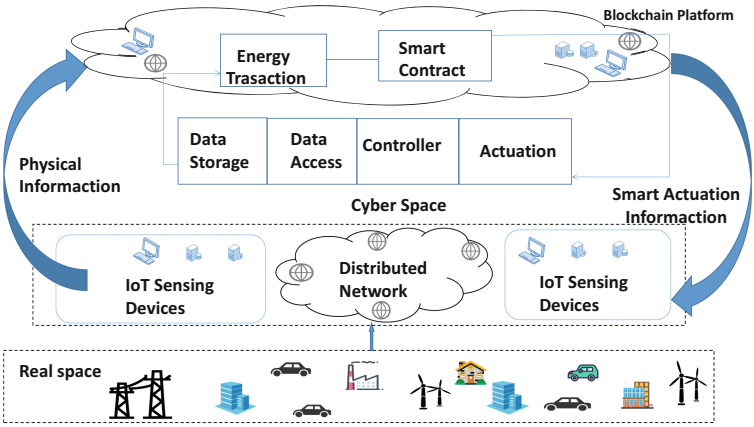



**Fig. 1.** Blockchain-enabled CPS.

### 2.1 Real Space Layer

The real space layer is the microgrid energy device. The data obtained from it is stored in Ethereum, and the actuation of the sensor is done through smart contracts. The main components of the real space layer include the grid, wind turbines, energy vehicles and residential users. We denote the set of $\mathcal{N} = \{1, ..., N\}$ residential users, as shown in Fig. 1. The residential users also own small renewable generators and energy storage units. The user model and the microgrid energy trading platform are described below.

1) User model: As shown in Fig. 1, the residential users and energy vehicles are used as user model. The user model consists of a load model, a power supply model and a battery storage model [14]. Furthermore, user $i \in \mathcal{N}$ can form a trading pair with user $j \in \mathcal{N}$ to exchange energy.
2) Supply Model: Wind power, energy vehicles and residential users in Fig. 1 can act as suppliers when they have an energy surplus. When the energy supplier in the system has no remaining energy to supply, the grid can be used as the energy supplier.
3) Battery model: All three of wind turbines, energy vehicles and residential users in Fig. 1 have a battery unit to charge the collected renewable energy or grid power, and then discharge it to provide a load [17].

### 2.2 Cyber Space Layer

In the cyber space layer, Internet-of-Things (IoT) sensing device sends the physical system to the blockchain layer, after receiving the information of the energy trading nodes, the IoT sensor device interacts with the distributed network, processes the data, and finally interacts with the blockchain layer.

### 2.3 Blockchain Layer

The blockchain layer feeds back the smart actuation information of the IoT sensor device. The blockchain platform focuses on the modeling of energy exchanges, and payments between users [18]. Blockchain and CPS integration can provide energy exchanges that require intelligent trading algorithms and information exchange platforms [19]. The energy transaction between users is carried out on the distribution network, and the payment is made using the blockchain token [20]. Finally, the final transaction is recorded on the blockchain smart contract.

## 3 PoWPB

The PoWPB considers the selection preference of nodes, and sets corresponding weights to the three components of TPB. At the same time, when selecting a high-credit authorization node, the vulnerability analysis is also taken into consideration, which solves the problems of reliability and low system performance.

### 3.1 TPB

The TPB [21] believes that human behavior is the result of a well-thought-out plan. Three variables determine the final behavioral intention (the inclination of an individual to take a certain action): behavioral attitude, subjective norm and perceived behavioral control. A detailed explanation of the factors is as fol lows:

1) Behavioral attitude: A positive or negative evaluation of an individual's performance of a particular behavior. e.g., an energy node knows that the counterparty it trades with has many previous good transaction records, and its behavioral attitude tends to trade with it.
2) Subjective norm: Refers to the perception of social pressure that an individual feels when he takes a certain behavior. e.g., a node sees better the evaluation, the more willing to trade with it.
3) Perceived behavioral control: The degree to which an individual expects to feel in control or mastery when taking a particular behavior. e.g., energy nodes will look at the balance in their account before trading. If the balance is sufficient, they will want to trade more.

### 3.2 Design of PoWPB

We designed the PoWPB consensus protocol based on the TPB. PoWPB takes into account the selection preferences of nodes, setting high-credibility nodes has a higher probability of becoming a candidate node of the authorization group. Combined with the Jain fairness index to comprehensively measure the trust and decentralization of the system, the system performance is finally optimized.

A linear combination model of individual behavioral is expressed as

$$B_i = \alpha A_i + \beta C_i + \gamma P_i, \tag{1}$$

where $B_i$ represents the ultimate behavioral intention of the individual, $\alpha, \beta, \gamma$ are the previous weight coefficients of $A_i, C_i, P_i$.

1) The aspiration $A_i$ here is equivalent to the trading willingness of a single node in the energy trading scenario. The incentive degree of nodes in individual transaction willingness comes from the subjective comparative feeling of the ratio of reward and investment between oneself and the reference object. That is, the nodes in the energy transaction will decide whether to trade with the matching energy production node according to the value of $A_i$ determined by the result calculated by the transaction willingness formula. Assuming that the average intensity of energy transactions is $\eta_i$ occurs during each block and assuming that the average energy transaction volume is $e_i$, then we have

$$A_i = \frac{R_i}{\rho \sum_{i=1}^{n} \eta_i e_i}, \tag{2}$$

where $R_i = \rho \frac{\sum_{i=1}^{n} \eta_i e_i}{N_{\mathrm{AG}}}$ is the reward and $\rho$ is the parameter related to $R_i$. Meanwhile, $N_{\mathrm{AG}}$ represents the number of high credit authorization groups.

2) The credit value $C_i$ is obtained by the following formula. The credit system is comprehensively used to evaluate the trustworthiness and reliability of users in the blockchain network [22]. In the microgrid energy trading scenario, participants play the role of blockchain nodes because they have sufficient computing and storage resources [23]. The system evaluates the reputation of each participant based on the historical evaluation feedback from the initiator. For convenience, the credit scores of three assessments are set separately, namely, a positive assessment with a credit score of $C^+$, a medium assessment with a credit score of $C^0$, and a negative assessment with a credit score of $C^-$. They can be set to positive for credit rewards, negative for credit penalties and zero for medium assessment. Considering the complexity of CPS, we introduced a CPS impact factor $\tau$. If the current latest block is the $k^{\mathrm{th}}$, then the reputation of the participant $i$ can be calculated from the latest $K$ block:

$$C_i = \tau \sum_{t=k-K+1}^{k} \sum_{j=1}^{C_i^t} C_{ji}^t, \tag{3}$$

where $C_{ji}^t$ represents is the credit evaluation set for energy node $i$ in $t^{\mathrm{th}}$ data block, $C_i^t$ is the $j^{\mathrm{th}}$ credit evaluation, $\tau$ is the impact factor of CPS. Equation (3) is the definition of system reliability.

3) With the development of modernized CPS for power grids, vulnerability assessment has become an emerging hotspot in power system security research [24]. Therefore, we introduce vulnerable node analysis. Vulnerable nodes need to be equipped with more powerful computing power to counter intrusion. The greater the computing power, the less difficult it is to mine or participate in consensus, and the nodes are more inclined to participate in transactions.

For a complex network $G = (V, E)$, where $V$ and $E$ represent the node set and the edge set, respectively, the above vulnerability indicators are defined as follows. The vulnerability of nodes can be assessed from two aspects of grid topology and operational status. In this paper, we introduce degree, betweenness and closeness centrality to quantify the topology vulnerability. State vulnerability is defined as the node load and electrical connection efficiency are used to quantify the vulnerability of grid nodes to study the impact of operating states on the physical changes of the power system. The comprehensive vulnerability index is the sum of topology vulnerability and state vulnerability. At the same time, Eq. (4) is also a measure of system vulnerability.

$$TS(i) = \omega \frac{T(i)}{\sum_{j \in V} T(j)} + (1 - \omega) \frac{S(i)}{\sum_{j \in V} S(j)}, \tag{4}$$

where $T(i)$ and $S(i)$ represent the structural vulnerability index and state vulnerability index, respectively, and $\omega$ is the weighting coefficient, which represents the proportion of topology information and state information in evaluating node vulnerability. Therefore, the expression for computing power $P_i$ is $P_i = TS(i)$.

### 3.3 Optimal Credit Threshold and Authorized Candidate Algorithm Design

The values of $A_i, C_i, P_i$ are guaranteed to be $0 \leq B_i \leq 1$. From a statistical point of view, the number of applicants (denoted as $N_{\mathrm{AP}}$ ) given by

$$\begin{aligned} N_{\mathrm{AP}} = \sum_{i=1}^{N} B_i = \sum_{i=1}^{N} (\alpha \frac{R_i}{\rho \sum_{i=1}^{n} \alpha_i e_i} + \beta \sum_{t=k-K+1}^{k} \sum_{j=1}^{C_i^t} C_{ji}^t + \\ \gamma(\omega \frac{T(i)}{\sum_{j \in V} T(j)} + (1 - \omega) \frac{S(i)}{\sum_{j \in V} S(j)})). \end{aligned} \tag{5}$$

**Theorem 1.** *Based on the consensus protocol modeled by TPB, the configurable credit threshold $\varepsilon$ for the candidacy of the authorizer can be determined from the normal probability distribution function.*

$$P\{C_i > \varepsilon\} = \frac{N_{\mathrm{CA}}}{N_{\mathrm{AP}}} = P\{C_i > \varepsilon\} = \int_{\varepsilon}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{6}$$

where

$$\mu = K(\eta^{+} C^{+} + \eta^{0} C^{0} + \eta^{-} C^{-}), \tag{7}$$

$$\sigma^2 = k^2(\sigma^{+^2}C^{+^2} + \sigma^{0^2}C^{0^2} + \sigma^{-^2}C^{-^2}). \tag{8}$$

The proof of Theorem 1 will be given in the appendix.

**Theorem 2.** *Considering Jain's fairness measure, the optimal number of candidate nodes and the credit threshold can be determined by the following optimization problem.*

**Algorithm 1.** Determine the optimal number of candidates $N_{\mathrm{CA}}$ and credit thresholds $\varepsilon$

**Initialize:** n=$N_{\mathrm{AG}}$ , $(\mathcal{F}_{jain})$max=0
**Input:** $\{C_i | i = 1, 2, ..., N_{\mathrm{AP}}\}$
**Output:** $N_{\mathrm{CA}}{}'$ , $\varepsilon'$
  Sort nodes according to $C_i$ credit score from high to low;
  Use Eq. (10) to calculate the average credit value $\overline{C}$ ;
  **for** $N_{\mathrm{CA}} = N_{\mathrm{AG}}$ to $N_{\mathrm{AP}}$ **do**
    The value $\varepsilon$ calculated from Eq. (6) ;
    Solve $\mathcal{C}$ from Eq. (9);
    Solve $\mathcal{D}$ from Eq. (11);
    The value $\mathcal{F}_{jain}$ calculated by Eq. (12);
    **if** $(\mathcal{F}_{jain})\mathrm{max} < \mathcal{F}_{jain}$ **then**
      $(\mathcal{F}_{jain})\mathrm{max} = \mathcal{F}_{jain}$ ;
      $N_{\mathrm{CA}}{}' = N_{\mathrm{CA}}$ ;
      $\varepsilon' = \varepsilon$ ;
    **end if**
  **end for**
  **return** $N_{\mathrm{CA}}{}'$ , $\varepsilon'$.

In the PoWPB consensus protocol, there are two conflicting considerations: on the one hand, high-reputation participants are expected to act as verifiers to ensure the security and credibility of the protocol. On the other hand, for decentralization considerations, all users should have the same opportunity to become a validator. Therefore, it is a difficult problem to comprehensively measure the degree of trust and decentralization of the consensus algorithm. Define trust $\mathcal{C}$:

$$\mathcal{C} = \frac{\varepsilon - \bar{C}}{\bar{C}}, \tag{9}$$

$$\bar{C} = \frac{\sum_{i=1}^{N_{\mathrm{AP}}} c_i}{N_{\mathrm{AP}}}, \tag{10}$$

where $\bar{C}$ represents the average evaluation reputation of the applicant. Define the degree of decentralization $\mathcal{D}$:

$$\mathcal{D} = \frac{N_{\mathrm{CA}}}{N_{\mathrm{AP}}}. \tag{11}$$

It represents the degree of distribution of block validators implemented by the consensus mechanism, where $N_{\mathrm{CA}}$ is a high-credit node or an authorizer candidate node and $N_{\mathrm{AP}}$ is the number of authorized candidate application nodes. In the PoWPB protocol, there is $N_{\mathrm{AG}}$ authorizer among $N$ nodes that are organized and stable. Subsequently, it is expected to achieve a good trade-off between the trust degree and the degree of decentralization of the consensus. Using Jain's fairness theory can be equivalently solved the following problem:

$$\begin{aligned} \mathcal{F}_{jain} &= \max_{N_{\mathrm{CA}},\varepsilon} \frac{(\sum_{i=1}^{N} \frac{\mathcal{C}}{\mathcal{D}})^2}{n \sum_{i=1}^{N} (\frac{\mathcal{C}}{\mathcal{D}})^2} \\ \text{s.t.} \quad & N_{\mathrm{AG}} \leq N_{\mathrm{CA}} \leq N_{\mathrm{AP}}. \end{aligned} \tag{12}$$

The optimal number of candidates $N_{\mathrm{CA}}$ and credit threshold $\varepsilon$ can be determined by Algorithm 1. So far, PoWPB has found the optimal number of validator candidates $N_{\mathrm{CA}}$ and the corresponding reputation threshold $\varepsilon$, and completed the selection of the consensus algorithm verification committee. The time complexity of Algorithm 1 is O(n), the time complexity of our designed algorithm 1 is still relatively low.

## 4 Numerical Tests

In this section, we briefly compare the proposed PoWPB with PoW and PoPB. In order to verify the reliability of the consensus mechanism we designed, we verified the node credit evaluation mechanism to test the impact of long-term goodwill transactions on the node consensus credit value. The user's microgrid network, and it is assumed that each node has the same electricity equity. It can be seen from Fig. 2, after multiple consensuses, the growth rate of the total consensus credit value approaches zero, and its value does not increase any more, which proves that the consensus mechanism PoWPB we designed can reasonably control the growth of node credit value, and there will be no single energy node overrun high, which threatens network security.

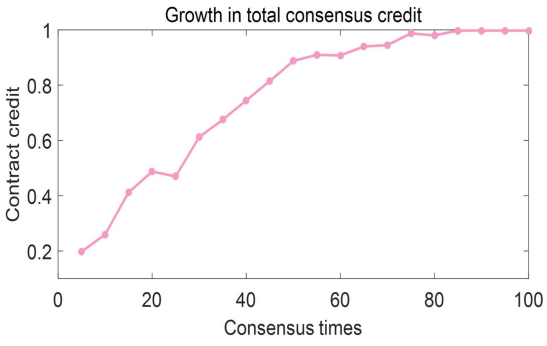
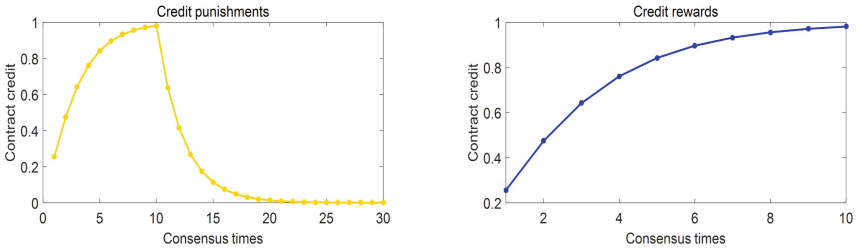



**Fig. 2.** Growth in total consensus credit.

For the purpose of more intuitively to see the rise and fall of the consensus credit value after a node conducts a good faith transaction and a malicious transaction, two scenarios are set up: in scenario 1, a consensus participant with good credit continuously votes maliciously, as shown in Fig. 3(a); scenario 2, the nodes continuously vote in good faith, as shown in Fig. 3(b). Figure 3 shows that the node contract credit value evaluation mechanism can effectively complete the evaluation of the node contract credit value. In Fig. 3(a), when the node conducts malicious transactions, the credit value of the node contract will drop rapidly. In Fig. 3(b), when a node conducts honest transactions, its node contract credit value can grow slowly.

Then, we implement three sets of experiments. First, the aim of the experiment is to compare the fairness of the PoWPB consensus with the PoPB consensus. Considering the different selection preferences of each node, in order to compare, we obtained four forms of PoPB algorithms by adjusting the final behavioral intention combination model of the node $B_i = \alpha A_i + \beta C_i + \gamma P_i$ the weight of each parameter, respectively. (i) Without considering the weight of each factor ($\alpha, \beta, \gamma$ all are 1). (ii) Without consider personal wishes $A_i$ ($\alpha = 0, \beta, \gamma$ both are 1). (iii) Without consider the reputation value $C_i$ ($\beta = 0, \alpha, \gamma$ both are 1). (iv) Without consider the computing power $P_i$($\gamma = 0, \alpha, \beta$ both are 1). They are denoted as PoPB-1, PoPB-2, PoPB-3 and PoPB-4 respectively.




(a) Credit punishments. (b) Credit rewards.

**Fig. 3.** Changes in credit value in different scenarios.

In Fig. 4, we can find that when the number of nodes in the network is less than 50, the fairness comparison between PoWPB and the four PoPB algorithms is not obvious, and when the number of nodes is greater than 50, the PoWPB algorithm is in the leading position, and when the number of nodes is greater than 100, the fairness of the four PoPB algorithms decreases, while the PoWPB algorithm tends to be stable and much larger than other algorithms. This is because when the number of nodes reaches a certain scale, the preference of nodes as the number increases becomes more and more obvious the influence on the number increasing becomes more and more obvious if the preference of nodes is not considered.

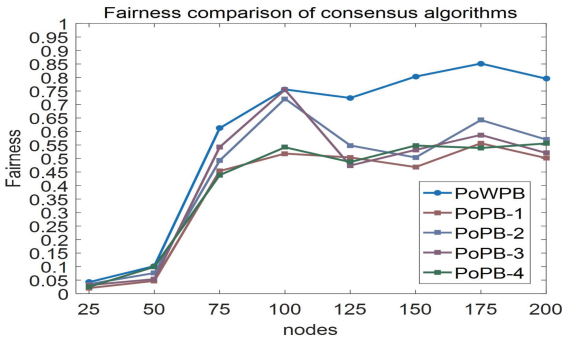
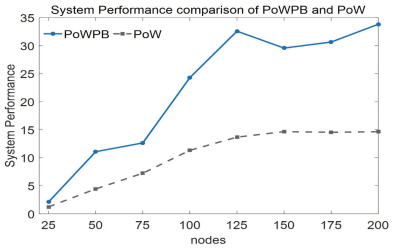
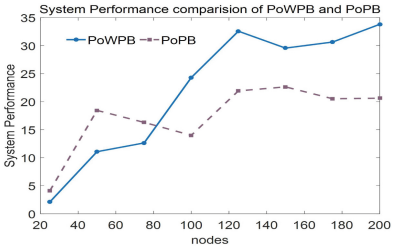



**Fig. 4.** Fairness comparison of consensus algorithms.

We compared the PoWPB consensus with the PoW consensus and PoPB consensus in terms of system performance. The system performance is related to the credibility $\mathcal{C}$ and decentralization $\mathcal{D}$ of the system. We define the system performance: $\varphi\mathcal{C} + (1-\varphi)\mathcal{D}$, where $\varphi$ is the parameter. As shown in Fig. 5(a) and Fig. 5(b), we can clearly see that the simulation results show that when the number of nodes in the network is the same, this paper proposes the PoWPB mechanism has a certain improvement in system performance. When the number of nodes increases, the advantages are more obvious.




(a) system performance comparision of PoWPB and PoW.




(b) system performance comparision of PoWPB and PoPB.

**Fig. 5.** System performance comparision.

In Fig. 5(a) and Fig. 5(b), the system performance of PoWPB with PoW and PoPB algorithms are compared respectively. It is also found that when the number of nodes is small, the advantages of PoWPB are not obvious. However, when the number of nodes increases, the system performance of PoWPB increases significantly. Finally, it decreases slightly and eventually stabilizes. This is because when the number of nodes in the network increases, the transaction volume in the system increases, and the reputation value accumulated by each node is more convincing and representative.

## 5 Conclusions

To solve the reliability issues and the low system performance of the integration of CPS and blockchain, we have designed a PoWPB consensus protocol. PoWPB is based on TPB to select high-credit nodes as consensus authorizers through a special authorizer group mechanism. We have taken the microgrid energy transaction integrated with blockchain and CPS as an example. Overall, our experimental results have demonstrated a strong effect that PoWPB can significantly improve the reliability of energy trading and the system performance. In the future, we will explore more efficient consensus mechanism to study the reliability and system performance of CPS based on the blockchain.

**Acknowledgements.** This work is partly supported by the National Natural Science Foundation of China (Nos. 62103375, 61877055), and the Zhejiang Provincial Natural Science Foundation of China (Nos. LY22F030006, 22NDJC009Z).

## A Appendix

### A.1 Proof of Theorem 1

Considering three credit evaluations, assume $\eta_i$ the hypothetical part $\eta_i^+$ is given a positive credit evaluation, $\eta_i$ the other part $\eta_i^0$ is given a moderate credit evaluation, and $\eta_i$ the other part $\eta_i^-$ is given a negative credit evaluation. Further, assume that the distribution of credit assessments is random, with $\{\eta_i^+\}_{i=1}^N \sim N(\eta_i^+, \sigma^{+2})$,$\{\eta_i^0\}_{i=1}^N \sim N(\eta_i^0, \sigma^{0^2})$,$\{\eta_i^-\}_{i=1}^N \sim N(\eta_i-, \sigma^{-^2})$. Then, the expectation of $C_i$ reference ([3]) can be given by

$$\bar{C}_i = K(\eta_i^+ C^+ + \eta_i^+ C^0 + \eta_i^+ C^-). \tag{13}$$

The distribution of $\{\bar{C}_i\}_{i=1}^N$ can be obtained by $\{\bar{C}_i\}_{i=1}^N \sim N(\mu, \sigma^2)$, where $\mu$ and $\sigma^2$ are defined by Eq. (7) and Eq. (8), respectively. Using $\bar{C}_i$ in Eq. (13) instead of $C_i$ in Eq. (6), the normal probability distribution function can be derived by Eq. (6). Finally, the credit threshold $\varepsilon$ can be solved by Eq. (6).

## References

1. Zhou, Z., Wang, B., Dong, M., Ota, K.: Secure and efficient vehicle-to-grid energy trading in cyber physical systems: Integration of blockchain and edge computing. IEEE Trans. Syst. Man Cybern. Syst. **50**(1), 43–57 (2020)
2. Skowronski, R.: The open blockchain-aided multi-agent symbiotic cyber-physical systems. Futur. Gener. Comput. Syst. **94**, 430–443 (2019)
3. Yang, Q., Wang, H.: Blockchain-empowered socially optimal transactive energy system: framework and implementation. IEEE Trans. Industr. Inf. **17**(5), 3122–3132 (2021)
4. Liu, K., Chen, W., Zheng, Z., Li, Z., Liang, W.: A novel debt-credit mechanism for blockchain-based data-trading in internet of vehicles. IEEE Internet Things J. **6**(5), 9098–9111 (2019)

5. Kang, J., Rong, Y., Huang, X., Maharjan, S., Yan, Z., Hossain, E.: Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains. IEEE Trans. Industr. Inf. **13**(6), 3154–3164 (2017)
6. Xiong, Z., Kang, J., Niyato, D., Wang, P., Poor, H.V.: Cloud/edge computing service management in blockchain networks: Multi-leader multi-follower game-based admm for pricing. IEEE Trans. Serv. Comput. **13**(2), 356–367 (2020)
7. Viriyasitavat, W., Xu, L.D., Bi, Z., Sapsomboon, A.: New blockchain-based architecture for service interoperations in Internet of Things. IEEE Trans. Comput. Social Syst. **6**(4), 739–748 (2019)
8. Guo, J., Ding, X., Wu, W.: A blockchain-enabled ecosystem for distributed electricity trading in smart city. IEEE Internet Things J. **8**(3), 2040–2050 (2021)
9. Li, Z., Kang, J., Yu, R., Ye, D., Deng, Q., Zhang, Y.: Consortium blockchain for secure energy trading in industrial internet of things. IEEE Trans. Industr. Inf. **14**(8), 3690–3700 (2018)
10. Chen, C., Wu, J., Lin, H., Chen, W., Zheng, Z.: A secure and efficient blockchain-based data trading approach for internet of vehicles. IEEE Trans. Veh. Technol. **68**(9), 9110–9121 (2019)
11. Zou, J., Ye, B., Qu, L., Yan, W., Lei, L.: A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services. IEEE Trans. Serv. Comput. **12**(3), 429–445 (2019)
12. Kim, J., Lee, J., Park, S., Choi, J.K.: Battery-wear-model-based energy trading in electric vehicles: A naive auction model and a market analysis. IEEE Trans. Industr. Inf. **15**(7), 4140–4151 (2019)
13. Yang, T., Zhai, F., Liu, J., Wang, M., Pen, H.: Self-organized cyber physical power system blockchain architecture and protocol. Int. J. Distributed Sensor Netw. **14**(10) (2018)
14. Wang, M., Xu, C., Chen, X., Zhong, L., Wu, Z., Wu, D.O.: Bc-mobile device cloud: A blockchain-based decentralized truthful framework for mobile device cloud. IEEE Trans. Industr. Inf. **17**(2), 1208–1219 (2021)
15. Li, W., Feng, C., Zhang, L., Xu, H., Cao, B., Imran, M.A.: A scalable multi-layer pbft consensus for blockchain. IEEE Trans. Parallel Distrib. Syst. **32**(5), 1146–1160 (2021)
16. Lin, F., Xia, S., Qi, J., Tang, C., Zheng, Z., Yu, X.: A parking sharing network over blockchain with proof-of-planned-behavior consensus protocol. IEEE Trans. Veh. Technol. **71**, 8124–8136 (2022). https://doi.org/10.1109/TVT.2022.3173989
17. Mollah, M.B., et al.: Blockchain for future smart grid: a comprehensive survey. IEEE Internet Things J. **8**(1), 18–43 (2021)
18. Li, M., Hu, D., Lal, C., Conti, M., Zhang, Z.: Blockchain-enabled secure energy trading with verifiable fairness in industrial Internet of Things. IEEE Trans. Industr. Inf. **16**(10), 6564–6574 (2020)
19. AlAshery, M.K., et al.: A blockchain-enabled multi-settlement quasi-ideal peer-to-peer trading framework. IEEE Trans. Smart Grid **12**(1), 885–896 (2021)
20. Hamouda, M.R., Nassar, M.E., Salama, M.M.A.: A novel energy trading framework using adapted blockchain technology. IEEE Trans. Smart Grid **12**(3), 2165–2175 (2021)
21. Ajzen, I.: The theory of planned behavior. Organ. Beh. Human Decis. Processes **50**(2), 179–211 (1991)
22. Guo, S., Hu, X., Guo, S., Qiu, X., Qi, F.: Blockchain meets edge computing: a distributed and trusted authentication system. IEEE Trans. Industr. Inf. **16**(3), 1972–1983 (2020)

23. Sun, G., Dai, M., Zhang, F., Yu, H., Du, X., Guizani, M.: Blockchain-enhanced high-confidence energy sharing in internet of electric vehicles. IEEE Internet Things J. **7**(9), 7868–7882 (2020)
24. Rahimi, S., Zargham, M.: Vulnerability scrying method for software vulnerability discovery prediction without a vulnerability database. IEEE Trans. Reliab. **62**(2), 395–407 (2013)

# E$^2$M: Evolving Mobility Modeling in Metropolitan-Scale Electric Taxi Systems

Yizong Wang, Haoyu Wang, Dong Zhao, Fuyu Yang, and Huadong Ma(✉)

Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China
{wangyizong,wanghaoyu2,dzhao,xiaoyuyfy,mhd}@bupt.edu.cn

**Abstract.** Human mobility data play an important role in addressing various urban issues. However, when a new mobility paradigm emerges and continuously evolves with time, it is usually hard to obtain a large-scale and evolving mobility dataset due to various factors such as social and privacy concerns. In this paper, we focus on modeling the evolving mobility of metropolitan-scale electric taxis (ETs), which have different mobility patterns with petroleum vehicles and continuously evolve with the expansion of the ET fleet and the charging station network. To this end, the E$^2$M system is proposed to generate trajectories for large-scale ET fleets by learning the mobility from only a small-scale ET fleet and the corresponding charging station network. *First*, the ET mobility is decomposed and modeled with transition, charging, and resting patterns. *Second*, the E$^2$M system generates trajectories with a fleet generation algorithm. Extensive experiments are conducted on a real-world dataset, which has ET trajectories during both the early stage and mature stage in the taxi electrification process in Shenzhen, China, and the results verify the effectiveness of E$^2$M.

## 1 Introduction

Human mobility data (*e.g.*, taxi trajectories) play an important role in addressing various related issues, *e.g.*, urban planning [1], disease control [2], and cellular traffic prediction [3]. In the recent decade, we have witnessed various emerging mobility paradigms in recent years, *e.g.*, shared bicycles [4], and electric vehicles [5], which exhibit novel mobility characteristics respectively. Especially, Electric Taxi (ET) contributes to significant emission reduction for better air quality and less energy consumption, which provides a higher incentive for city governments to upgrade conventional gas taxis with ETs. Therefore, we focus on the ET mobility modeling.

When a new mobility paradigm emerges and continuously evolves with time, it is usually hard to obtain a large-scale and evolving mobility dataset due to various factors such as social and privacy concerns. More specifically, the inevitable taxi electrification has been accomplished in a few cities, *e.g.*, Shenzhen, China


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 126–138, 2022.
https://doi.org/10.1007/978-3-031-19208-1_11

[6]. By contrast, taxi electrification has not started (or just started) in many cities, where the scarcity of ET mobility data limits consequential applications. Therefore, *it is of great value to model the metropolitan-scale ET mobility, which exhibits unique mobility patterns and evolves with the expansion of the ET fleet and charging station network.*

The research on traditional human mobility models has gone through four stages: *1)* Synthetic context-free models [7] generate mobile trajectories according to some predefined regulations. *2)* Synthetic context-aware models [8] take obstacles or road topologies into consideration. *3)* Data-driven context-free models [9,10] incorporate statistical features from observed data. *4)* Data-driven context-aware models [2,11] leverage both statistical features and contexts. The traditional synthetic mobility models are unsuitable for the emerging mobility pattern of ET which has a shorter mileage and the unique charging behavior. The data-driven models have ability to capture more complex mobility pattern by learning it from the sensor data. However, the data-driven models compromise on modeling the emerging ET mobility for two reasons: *1)* the scarcity of ET sensor data hinders the mobility modeling. *2)* current models [2,11] "replay" the mobility from the datasets with static context, which is not applicable to the continuously evolving context, *i.e.*, the expanding ET fleet and charging station network. When the direct sensor data for the emerging electric vehicles are not accessible, the existing works usually leverage implicit data, *e.g.*, population density [5] and data of petroleum taxis and stations [12], to speculate the mobility. However, this way is error-prone. Although Wang et al. [6] investigate the long-term evolving patterns of ET networks by measurements, they have not addressed how to model and predict such evolving patterns.

In this paper, we propose a generative mobility model, which can learn the mobility patterns from a small-scale ET fleet and the corresponding charging station network, and then generate trajectories for varied numbers of ETs and varied distributions of charging stations. Such a generative model of evolving mobility has three significant merits. *First, it has better availability.* The generative mobility model can provide mobility data in a publicly available manner without privacy concerns. *Second, it has better flexibility.* The generative model of evolving mobility can support what-if study, e.g., charging station siting and charging recommendation, because it can generate trajectories with a varied scale of ET fleets and varied distribution of charging stations. *Third, it has better compactness.* A one-month anonymized ET trajectory dataset for a metropolitan area occupies approximately 300 gigabytes. By contrast, the generative mobility model can be stored as model parameters within 1 gigabyte.

It is challenging to model the evolving mobility patterns with the complex dynamics: *1)* charging network dynamics, i.e., new open or closed charging stations will affect other stations and the ET mobility; *2)* ET network dynamics, i.e., some key metrics (e.g., waiting time for charging, charging station utilization rate) have a complex and non-linear relationship with the number of ETs; *3)* contextual factor dynamics, i.e., multi-variant impact factors, such as time of day and day of the week, affect the ET mobility pattern.

To this end, we present the Evolving ET Mobility ($E^2M$) system to generate ET trajectories at a metropolitan scale. The $E^2M$ system models the ET mobility

by modeling the transition, charging, and resting patterns respectively. For a specific charging station distribution, the $E^2M$ system generates trajectories for an ET with a deterministic finite automaton algorithm, which integrates the three modeled patterns. Considering the queuing procedure among ETs, we further propose a fleet generation algorithm to generate trajectories in parallel for an ET fleet. In summary, our contributions are listed as follows:

- We conduct a work to model the evolving mobility in metropolitan-scale ET systems. It differs from the existing works that model general human mobility or statistically replay the observed mobility.
- We propose a novel $E^2M$ system which can generate ET trajectories for varied scale of ET fleet and varied CS distribution. It shows three significant merits: better availability, flexibility, and compactness.
- Extensive experiments on a three-year metropolitan-scale ET trajectory dataset in Shenzhen, China, demonstrate the advantages of $E^2M$ over the baselines.
- We have released an open-source trajectory generation service[1] for public use. Specifically, it takes the number of ETs and the charging station distribution as input and generates publicly available trajectories for potential research and applications.

## 2 Overview

### 2.1 ET Contextualization

**Definition 1 (Raw ET Trajectory).** *A raw ET trajectory collected from an individual ET is essentially a sequence of tuples* $(t, l)$*, which indicates an ET in location* $l$ *at the timestamp* $t$*.*

**Definition 2 (ET Event).** *Each event* $e_i$ *is represented by a tuple* $(c, t, l, d)$*, where the elements are category, timestamp, location, and duration of the event, respectively. Four categories of events are considered, which are Pick-up, Drop-off, Charging, and Rest events, i.e.* $c \in \{\text{PE}, \text{DE}, \text{CE}, \text{RE}\}$*. Note that* $d = 0$ *if* $c \in \{\text{PE}, \text{DE}\}$*, because pick-up and drop-off events are instant events.*

**Definition 3 (Semantic ET Trajectory).** *We use a series of events* $E = \{e_1, \cdots, e_n\}$ *to semantically represent the trajectory of an ET.*

Following [6], we use Fig. 1 to illustrate the transition, charging, and rest patterns of ETs in the context comprising three dimensions: spatial, temporal, and energy:

**Transition Pattern**. The ET seeks for the first passenger after fully charged at $t_0$, consuming time $t_{\text{picking}} = t_1 - t_0$, till that a pick-up event $e_i = (\text{PE}, t_1, \text{Pick-up}_1, 0)$ happens. The ET will drop-off the passenger as required, where a

[1] https://github.com/easysam/electric-taxi-mobility.

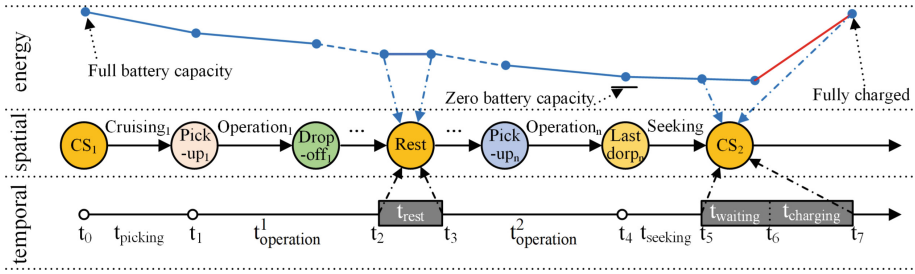



**Fig. 1.** The operation, charging, and rest patterns of ET.

drop-off event $e_{i+1} = (\text{DE}, t_{i+1}, \text{Drop-off}_1, 0)$ happens. Given any pick-up location $l_i$, the observed distribution among destinations $l_{i+1}$ of passengers reflects the overall transition demand from $l_i$ to $l_{i+1}$. Given any drop-off location $l_j$, the observed distribution among next pick-up location $l_{j+1}$ reflects the seeking strategy of the taxi drivers at $l_j$. For simplicity, cruising process after recharging is considered the same with seeking next passenger after a drop-off event. Therefore, the consumed time for transition contains $t_{\text{picking}}$, $t^1_{\text{operation}}$, and $t^2_{\text{operation}}$, during which the state of charge (SOC) is reducing.

**Charging Pattern**. A driver will decide whether to recharge the ET after dropping off the passengers, and if yes, which charging station he will choose to charge. Then charging events happen in the charging stations chosen by the ET drivers. The time ahead of charging contains $t_{\text{seeking}}$, $t_{\text{waiting}}$, and $t_{\text{charging}}$ in the example. The battery is fully charged after a charging event.

**Rest Pattern**. A driver will decide whether to rest after dropping off the passengers. The ET rest events behave as continuous stay points in the raw trajectories. The rest events usually happen in the early morning when the travel demand is small. The SOC keeps constant during a rest.

### 2.2 Framework Overview

Figure 2 illustrates the framework of our E$^2$M system, where the work process comprises three cascaded stages.

**Data Preprocessing**. It converts the raw trajectory to ET events. More specifically, we cluster the pick-up and drop-off events in *Hotspot Detection*; In *Mobility Event Extraction*, raw trajectories are transformed into semantic trajectories comprising pick-up, drop-off, charging, and rest events; In *Map Matching*, we project trajectory points from raw coordinates to points in the road network.

**Mobility Modeling**. *Transition Pattern Modeling* learns the transition mobility evolving pattern from mobility events of ETs and context data. *Charging Pattern Modeling* learns the ET charging behavior patterns. *Resting Pattern Modeling* leverages empirical probability distribution in several aspects to model the ET rest pattern.

**Trajectory Generation**. It generates semantic trajectories for a given number of ETs and the distribution of chargers. As an application, the utilization rate of charging stations can be calculated from the generated trajectories.

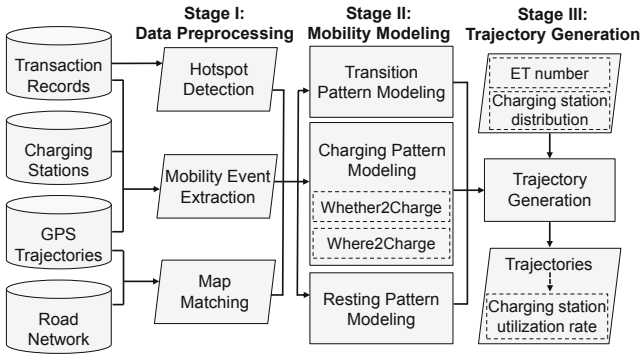



**Fig. 2.** Framework of the E$^2$M system.

## 3 Data Preprocessing

### 3.1 Mobility Event Extraction

Among the four kinds of ET events, pick-up events and drop-off events can be obtained from the taxi transaction data, but charging events and rest events are not directly available. Therefore, we extract charging and rest events from raw ET trajectory.

**Charging Event Extraction**: The charging events have unique characteristics, i.e., it happens spatially close to a charging station and temporally continues between half an hour and 2 h [13]. Based on these features, we design a spatio-temporal constraint based approach to extract charging events of ETs from their trajectories. The sub-trajectories are extracted based on the fact that an ET will stay within a specific range $r_i$ of the charging station $c_i$ for $\tau \in [0.5, +\infty)$ hours. To determine the range threshold $r_i$ of each charging station $c_i$, we manually label 100 charging events by plotting raw trajectories and locations of charging stations on Google satellite map. Each charging event comprises the timestamps of arriving at the CS, beginning charging, and leaving the CS. Then the range threshold $r_i$ for a charging station $c_i$ is determined as the maximum distance among the location of $c_i$ and the trajectory points within the labeled charging events happened at $c_i$.

**Rest Event Extraction**: Rest events have similar but simpler characteristics than charging events. The rest events are extracted based on the fact that an ET will stay for a long time at the same point to have a rest. Note that the RE extraction will be conducted after the charging event extraction to avoid detecting charging events as rest events.

### 3.2 Hotspot Detection

To model the preference distribution of spatially close ETs, we detect the drop-off hotspots by clustering the drop-off locations. The investigated region in Shenzhen is divided into $400 \times 800$ grids. Each drop-off event is mapped to a grid. We use MeanShift [14] to cluster the drop-off events. Because MeanShift is sensitive to noises, we filter out clusters with few drop-off events.

## 4 Mobility Modeling

### 4.1 Transition Pattern Modeling

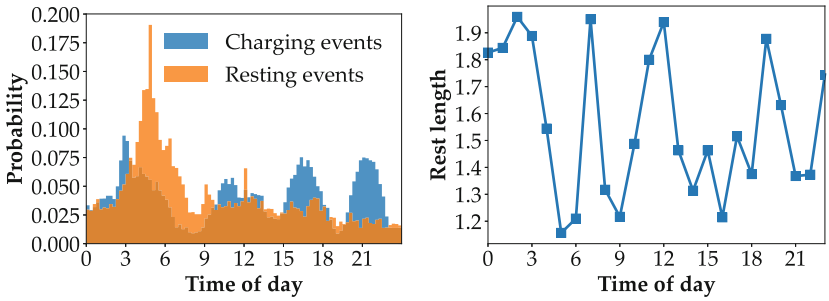
The transition pattern is described by the matrices of transition probability between grids. There are two transition types for ETs: from pick-up grids to drop-off grids (P2D), and the reverse (D2P). In both cases, the original grid and destination grid are denoted by O-grid and D-grid, respectively. We use the utility model [15] to learn the transition probability distribution between grids based on context features extracted for each pair of grids to model the transition patterns. Specifically, we extract a $(2+4k)$-dimension feature for each grid pair: mean travel time, mean travel distance, distances to $k$-nearest CSs of O-grid, charger numbers of $k$-nearest CSs of O-grid, distances to $k$-nearest CSs of D-grid, charger numbers of $k$-nearest CSs of D-grid.

The utility function assigns a utility score for each OD pair. The utility score represents the possibility of transition by ETs between the corresponding O- and D- grids. Then, a softmax function is applied for utility scores with the same O-grid to obtain a transition probability distribution from that O-grid to all possible D-grids. Specifically, the utility function is implemented as a multiplication of the ET willingness score and the observed transition number of the petroleum taxis. The ET willingness score is predicted based on the $(2+4k)$-dimension feature. The ET willingness score represents how ETs are willing to conduct the transition considering the abundance of charging resources near the O- and D- grids. The ET willingness score is predicted using the XGBRegressor and the ground truth is defined by the fraction of the observed number of ET transitions to that of taxi transitions.

### 4.2 Charging Pattern Modeling

Given that an ET just dropped the passenger(s), predicting where this ET will go to charge is the core problem in the modeling of charging pattern. The key idea is to decompose this problem into the following two problems: (1) Will this ET subsequently go to any charging station? (2) If yes, which charging station will this ET enter next? To answer these problems, we define two labels for each drop-off event.

**Definition 4 (WhetherToCharge Label).** *For an event $e_i$ with category $c_i =$ DE, its WhetherToCharge label is defined as* $z_i = \begin{cases} 1 & c_{i+1} = \text{CE} \\ 0 & c_{i+1} \in \{\text{PE}, \text{RE}\} \end{cases}$.




(a) Probability distribution of rest event among time of day.

(b) The average duration of rest events in each hour.

**Fig. 3.** Rest pattern of the ETs in 2014.

**Definition 5 (WhereToCharge Label).** *For an event $e_i$ with WhetherToCharge label $z_i = 1$, its WhereToCharge label is defined as $y_i$, which represents the index of the charging station where the charging event $e_{i+1}$ happens.*

With the labels $z_i$ and $y_i$ as the answers to the two problems, we build two models learned from historical data to predict the two labels for each drop-off event.

**WhetherToCharge Prediction.** The WhetherTocharge label prediction problem can be modeled as

$$P(z_i|F_i^A), \forall i, \text{where } e_i = DE \tag{1}$$

Prediction of $z_i$ is a binary classification problem. A 7-d feature $F_i^A$ is built for each drop-off event $e_i$, which is composed by time of day, whether weekday or not, traveled distance after last charging event, and four statistical values (min, max, median, mean) of distances among $l_i$ and all charging stations. We employ the classical classification models, XGBoost and multi-layer perceptron, as the prediction models.

**WhereToCharge Prediction.** To model the preferences for charging stations with varied distances and charger numbers at varied times. We aggregate WhereToCharge labels of drop-off events that happened in the same drop-off hotspot to a probability distribution $\boldsymbol{Y}$. We use a utility function based model to predict the distribution. For a CS, a 9-d feature is built by whether weekday or not, time of day, distance to the CS, charger number, four statistical values (min, max, median, mean) of distances to all CSs, and traveled distance after the last charging.

A 4-layer fully connected network $G_i$ is used to predict a score $q_i$ for each charging station $s_i$. The numbers of neurons in each layer of $G_i$ are 16, 32,

**Table 1.** Proportion of rest times in the ET fleet.

| Rest times | 0 | 1 | 2 | 3 | Others |
|---|---|---|---|---|---|
| Proportions | 0.1084 | 0.6611 | 0.2003 | 0.0226 | 0.0045 |

16, and 1. Batch normalization and relu function are added in each of the first three layers. A softmax function is applied on $\boldsymbol{q} = \{q_i | i = 1, \cdots, N\}$ to get the predicted $\hat{\boldsymbol{Y}}$. A KLDivLoss is used to measure the error, and SGD is used to optimize the model. The learnable weights are shared among $G_i$. Therefore, the utility function based model can adapt to any number of charging stations $N$, by building a $G_i$ for each charging station. This advantage addresses the problem that numbers of charging stations are different between observed and unseen periods.

### 4.3 Resting Pattern Modeling

The rest patterns are modeled by three statistics, which describe daily rest times, begin time of rests, and rest duration, respectively. The proportions of daily rest times of the ET fleet reflect the frequency of rest events. As shown in Table 1, most ETs rest once per day. Figure 3a illustrates the probability distribution of begin time of rest events. The peak located between 4 to 7 am are caused by the scarce of passengers and the rest of drivers at night. Figure 3b illustrates the average duration of the rest events that start at each hour of day, where four peaks are present. They may be caused by the low travel demand.

## 5 Trajectory Generation

### 5.1 Deterministic Finite Automaton Algorithm

As shown in Fig. 4, ET has four states, *i.e.*, empty, occupied, charging, and resting. The charging state also comprises queuing and charging processes. An arrow connecting two states indicates that the previous state can convert to the latter state via an event. More specifically, we generate state changes according to several rules: (1) Conversions from occupied to empty follow the pick-up to drop-off transition probability matrices. (2) Conversions from empty to occupied follow the drop-off to pick-up transition probability matrices. (3) Conversion from empty to charging is predicted by WhetherToCharge and WhereToCharge prediction models. (4) Conversion from empty to resting follows the resting patterns.

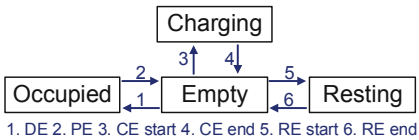



**Fig. 4.** The DFA of ET.

**Charging Time-Consuming**. The battery's remaining capacity can be estimated according to the traveled distance after the last fully charging event [16]. Specifically, the remaining capacity of the battery in % are $C_{t_2} = C_{t_1} - d \cdot u$, where $C_{t_1}$ is the max capacity of the battery (100%); $d$ is the traveled distance after last fully recharging; $u$ is the mean capacity consumption given in % per km. With $C_{t_2}$, we can estimate the time required to charge fully.

### 5.2 Fleet Generation Algorithm

Considering the queuing situation based on the first-come, first-served principle, generating a charging event will rely on all ETs arriving at the charging station earlier. Therefore, we record the time of the last arrival at any charging station for each ET as a vector $\boldsymbol{t} = \{t_1, \cdots, t_n\}$. Then the algorithm selects the ET with the earliest arrival time in $\boldsymbol{t}$ and generates the CE and following events until the ET arrives at any one charging station.

**Algorithm 1:** Fleet Generation

**input :** ET number $n$; charger distribution $D^C$;
ET initial distribution $D^L$; resting schedule $\boldsymbol{R}$
**output:** Generated trajectories $\boldsymbol{\Phi} = \{\Phi_i | i = 1, \cdots, n\}$

1 **for** $j \leftarrow 1, \ldots, |D^C|$ **do**
2  **for** $k \leftarrow 1, \ldots, D_j^C$ **do** $\boldsymbol{w}_k^j = 0$ ;
3 **for** $i \leftarrow 1, \ldots, n$ **do**
4  $\Phi_i, t_i, s_i, t_i^c \leftarrow \mathrm{DFA}(t_0, D_i^L)$; `// `$s_i$`, arrived CS; `$t_i^c$`, charging time`
5 $t_{\text{previous}} \leftarrow 0$;
6 **while** *true* **do**
7  $i \leftarrow \arg\min(\boldsymbol{t})$; `// The earliest arrived ET`
8  **if** $t_i \geq t_0 + \delta$ **then return**;
9  $\boldsymbol{w} \leftarrow \max(\boldsymbol{w} - (t_i - t_{\text{previous}}), 0)$ ; `// Updating` $\boldsymbol{w}$ `with passed time`
10  $k \leftarrow \arg\min(\boldsymbol{w}^{s_i})$ ; `// The earliest available charger in CS` $s_i$
11  $q \leftarrow w_k^{s_i}$; `// Queuing time-consuming`
12  $t_{\text{previous}} \leftarrow t_i + q$;
13  $w_k^{s_i} \leftarrow w_k^{s_i} + t_i^c$ ; `// Updating` $\boldsymbol{w}_k^{s_i}$ `with charging time` $t_i^c$
14  $\Phi_i$.append$(CE, t_i, l_i, q + t_i^c)$;
15  $\varphi, t_i, s_i, t_i^c \leftarrow \mathrm{DFA}(t_i + q, l_i)$
16  $\Phi_i$.extend$(\varphi)$;

This algorithm, shown in Algorithm 1, takes ET number, ET initial location distribution, charger distribution, and predefined resting schedule as its input. In this algorithm, $\boldsymbol{w}_k^j$, initialized to 0, represents the time interval after which the $j$-th charger in the $k$-th station will be available (line 1–2). The DFA algorithm generates trajectories for an ET, starting from the end of a charging event and ending at the beginning of the next charging event (the arrival at the CS).

The single period generations among ETs are independent and run in parallel (line 4 and 15). The while loop (line 7–16) selects the earliest ET arriving at any charging station (line 7) and generates a CE (line 14) as well as following events (line 15–16) for it. $\boldsymbol{w}$ is updated by subtracting the elapsed time since last generation of a CE, $t_i - t_{\text{pre}}$ (line 9–10). The fleet generation algorithm ends when the time has exceeded the preset generation duration (line 8).

## 6 Evaluation

### 6.1 Dataset and Experimental Settings

We use the ET trajectories dataset collected from Shenzhen, China in two time periods, named ET-2014 and ET-2017. The ET-2014, ranging from 3 July 2014 to 18 July 2014, comes from 664 ETs and 23 CSs, which is an early stage in ET networking development. The ET-2017, ranging from 1 June 2017 to 30 June 2017, comes from 10687 ETs and 564 CSs. The $E^2M$ system takes ET-2014 as input and generates ET trajectories with contexts of ET-2014 and ET-2017 respectively.

All the experiments are run in a Linux server (CPU: Intel(R) E5-2620 v4 @ 2.10GHz, Memory: 128 GB, GPU: NVIDIA Tesla P100). $k$ in transition pattern modeling is set to 3.

### 6.2 Transition Pattern Modeling Evaluation

**Table 2.** KL-divergences among transition probability distribution and different estimation methods

| Transition direction | ET data in 2014 | Taxi data in 2014 | Prediction |
|---|---|---|---|
| Pick-up → Drop-off | 20.83 | 0.2061 | 0.2032 |
| Drop-off → Pick-up | 11.97 | 0.2155 | 0.1083 |

Table 2 shows the results of transition prediction. Our $E^2M$ system gets the smallest KL-divergence with the ground truth transition matrix, compared to the ET and taxi fleet in the early stage i.e., July 2014. Transition probabilities of ET fleet in 2014 have a large KL-divergence with the target, which is caused by the expansion of ET fleet and charging network along time. Note that the proposed method reduces KL-divergence by 49.74% in the D2P transition prediction, compared to the taxi fleet in 2014, but there is only a 1.41% reduction in the P2D transition prediction. The reason may be that the P2D transition probability is mainly determined by passengers, which is similar to the traditional taxis. In contrast, the D2P transition probability of ET has more difference with the conventional taxis due to the consideration of charging resources around the destinations.

**Table 3.** Results of WhetherToCharge prediction

| Methods | Recall | Precision | Accuracy |
|---|---|---|---|
| XGBoost | 0.7223 | 0.2809 | 0.7304 |
| MLP | 0.3800 | 0.6500 | 0.9019 |

**Table 4.** Results of WhereToCharge prediction

| Methods | KLDiv |
|---|---|
| Real-time recommendation | 0.4848 |
| Even | 0.0687 |
| Observed data | 0.0625 |
| WhereToCharge | 0.0036 |

### 6.3 Charging Behavior Prediction

**WhetherToCharge Label Prediction.** As shown in Table 3, XGBoost predicts more true results with higher recall score, and MLP predicts less true results with higher precision score. Overall, we choose the MLP for its higher accuracy.

**WhereToCharge Label Prediction.** We use three baselines: (1) Real-time recommendation [13], which recommends a charging station for an ET; (2) Observed data, the observed visit distribution among CSs by all ETs; (3) Even, the uniform distribution among all charging stations. The observed data intuitively represents the preference of the whole ET fleet on CSs, but it does not consider the factors of drop-off locations.

As shown in Table 4, the proposed WhereToCharge label prediction method has the lowest KL-divergence. The real-time recommendation has a large KL-divergence because it returns a distribution where only the probability of returned charging station equals 1, and that of others equal 0, which is unsuitable to describe the charging station preference for ET fleets.

**Table 5.** Ranking result of usage frequency of charging station in generated trajectories.

| Method | HR@20 | NDCG@20 | HR@30 | NDCG@30 | HR@40 | NDCG@40 |
|---|---|---|---|---|---|---|
| ET data in 2014 | 0.1000 | 0.0286 | 0.1667 | 0.0529 | 0.2250 | 0.0751 |
| Taxi data in 2014 | 0.1500 | 0.0273 | 0.1667 | 0.0506 | 0.2750 | 0.0717 |
| $E^2M$ | **0.2000** | **0.2219** | **0.2667** | **0.2405** | **0.2750** | **0.2574** |

### 6.4 Generation Evaluation

Charging event is a important mobility pattern for ET and valuable information in electric vehicle expansion. Therefore, we evaluate evaluate the quality of generated trajectory by comparing the charging event distribution between the simulated mobility trajectory and the real mobility trajectory.

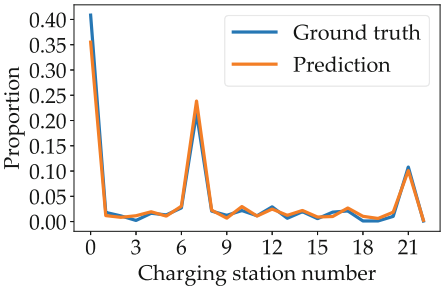



**Fig. 5.** Usage frequency distribution of charging stations.

**Mobility Replay**: We generate semantic trajectories for the ET fleet in 2014. We aggregate charging events in all charging stations to a distribution and compare it to the ground truth distribution. As shown in Fig. 5, the predicted distribution is closest to the ground truth with a KL divergence 0.0355. The result indicates that the $E^2M$ system can effectively replay the ET mobility that has similar charging station utilization rates with the observed utilization rates.

**Evolving Mobility Simulation**: We employ two metrics: Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) for quantitative evaluation. The larger the metric values are, the better the simulation results are. As shown in Table 5, our $E^2M$ system outperforms the other two baselines. The $E^2M$ system predicts the charging demand of each charging station with higher HR and NDCG result in all scale of $k$. The results verify that our $E^2M$ is able to model the evolving mobility and simulate the charging event distribution under varied contexts.

## 7 Conclusion

In this paper, we investigate an important problem of modeling the evolving mobility in metropolitan-scale electric taxi systems. A generative model E2M is proposed to model the ET mobility with varied scale of ET fleet and charging network. The E2M models the evolving mobility by modeling the transition, charging, and rest pattern. Then, it generates trajectories for large-scale ET fleet and charging station network, using a fleet generation algorithm. We evaluate E2M on a real-world dataset that contains trajectories of ET fleet in both early and mature stages in the taxi electrification process in Shenzhen, China, and the results verify the effectiveness of our E2M. This work is promising to benefit related research, e.g., charging station siting and charging recommendation.

**Acknowledgements.** This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101200; in part by the National Natural Science Foundation of China under Grant 61972044 and Grant 61732017; in part by the Fundamental Research Funds for the Central Universities

under Grant 2020XD-A09-3; in part by the Funds for International Cooperation and Exchange of NSFC under Grant 61720106007; and in part by the 111 Project under Grant B18008.

## References

1. Wu, G., Li, Y., Bao, J., Zheng, Y., Ye, J., Luo, J.: Human-centric urban transit evaluation and planning. In: IEEE ICDM, pp. 547–556 (2018)
2. Feng, J., Yang, Z., Xu, F., Yu, H., Wang, M., Li, Y.: Learning to simulate human mobility. In: ACM SIGKDD, pp. 3426–3433 (2020)
3. Wang, X., et al.: Spatio-temporal analysis and prediction of cellular traffic in metropolis. IEEE TMC **18**(9), 2190–2202 (2019)
4. Yang, Z., Hu, J., Shu, Y., Cheng, P., Chen, J., Moscibroda, T.: Mobility modeling and prediction in bike-sharing systems. In: MobiSys, pp. 165–178 (2015)
5. Xu, Y., Çolak, S., Kara, E.C., Moura, S.J., González, M.C.: Planning for electric vehicle needs by coupling charging profiles with urban mobility. Nat. Energy **3**(6), 484–493 (2018)
6. Wang, G., Chen, X., Zhang, F., Wang, Y., Zhang, D.: Experience: Understanding long-term evolving patterns of shared electric vehicle networks. In: MobiCom (2014)
7. Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y.-C., Jetcheva, J.: A performance comparison of multi-hop wireless ad hoc network routing protocols. In: ACM/IEEE MobiCom, pp. 85–97 (2022)
8. Jardosh, A., Belding-Royer, E.M., Almeroth, K.C., Suri, S.: Towards realistic mobility models for mobile ad hoc networks. In: MobiCom, pp. 217–229 (2003)
9. Rhee, I., Shin, M., Hong, S., Lee, K., Kim, S.J., Chong, S.: On the levy-walk nature of human mobility. IEEE/ACM TON **19**(3), 630–643 (2011)
10. Lee, K., Hong, S., Kim, S.J., Rhee, I., Chong, S.: Slaw: Self-similar least-action human walk. IEEE/ACM TON **20**(2), 515–529 (2011)
11. Kang, X., Liu, L., Zhao, D., Ma, H.: Trag: a trajectory generation technique for simulating urban crowd mobility. IEEE TII **17**(2), 820–829 (2020)
12. Liu, C., Deng, K., Li, C., Li, J., Li, Y., Luo, J.: The optimal distribution of electric-vehicle chargers across a city. In: IEEE ICDM, pp. 261–270 (2016)
13. Tian, Z., et al.: Real-time charging station recommendation system for electric-vehicle taxis. IEEE TITS **17**(11), 3098–3109 (2016)
14. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE TPAMI **24**(5), 603–619 (2002)
15. Prato, C.G., Bekhor, S.: Modeling route choice behavior: how relevant is the composition of choice set? TRR **2003**(1), 64–73 (2007)
16. Hess, A., Malandrino, F., Reinhardt, M.B., Casetti, C., Hummel, K.A., Barceló-Ordinas, J.M.: Optimal deployment of charging stations for electric vehicular networks. In: UrbaNe, pp. 1–6 (2012)

# Multi-task Class Feature Space Fusion Domain Adaptation Network for Thyroid Ultrasound Images: Research on Generalization of Smart Healthcare Systems

Xiang Ying[1,2,3], Zhen Liu[1,2,3], Jie Gao[1,2,3], Ruixuan Zhang[1,2,3], Han Jiang[4], and Xi Wei[5(✉)]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
[2] Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin, China
[3] Tianjin Key Laboratory of Advanced Networking, Tianjin, China
[4] OpenBayes (Tianjin) IT Co., Ltd., Tianjin, China
[5] Tianjin Medical University Cancer Institute and Hospital, Tianjin, China
weixi@tmu.edu.cn

**Abstract.** In recent years, the poor generalizability of deep neural networks in multi-model medical images has attracted widespread attention. Domain adaptation is an approach to alleviate the above problem, which transfers the labeled source domain to the target domain. It can reduce the data labeling workload in the target domain and significantly improve the network's generalizability. However, the differences between foreground areas and background areas of medical images are relatively minor, and it is difficult for existing methods to effectively extract domain invariant features. Further optimization of the feature distribution alignment for each category is also lacking. Therefore, a Multi-task Class feature space Fusion Domain Adaptation Network (MCFDAN) is proposed in this paper. Firstly, a reconstruction branch is added to the baseline network to mitigate feature offset of the target domain during encoding. Secondly, category constraints are added to the fusion of domain feature spaces, improving the generalizability of the source classifier to the target domain. Finally, the network incorporates a recurrent cross-attention module that highlights the feature expression of the lesion region. The evaluation results demonstrate that the proposed network achieves a significant performance improvement, which is important for the application of smart healthcare systems.

**Keywords:** Domain adaptation · Thyroid ultrasound images · Smart healthcare systems · Generalizability


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 139–152, 2022.
https://doi.org/10.1007/978-3-031-19208-1_12

## 1 Introduction

Deep learning based on neural networks has been widely used in computer vision tasks, such as image classification [7], image segmentation [3], and object detection [16]. In addition, the convolutional neural network has been increasingly used in smart healthcare systems to assist diagnosis, and has gradually become an important tool to analyze the lesion area in medical images. However, due to the differences in scanning methods and output frequencies of different medical imaging instruments, multiple domains are formed, and each domain has a specific semantic expression (such as image style, texture, and image quality), that is domain shift. As shown in Fig. 1, multi-model thyroid ultrasound images show domain shift features (such as texture features, artificial marker, etc.), and also show domain invariant features (such as edge, calcification, etc.) with consistent semantics. The existing deep neural networks are difficult to extract the weak domain invariant features of thyroid nodule regions between different domains. While supervised deep networks trained on the specific domain can achieve good results, their performance is often poor in similar target domains with consistent category space and inconsistent feature distribution. In order to solve the problem of domain shift, one of the most commonly used methods by researchers is domain adaptation [15]. By obtaining large-scale labeled data of a domain as the source domain, it can assist in learning relevant but unlabeled target domain data. This paper takes the multi-model thyroid ultrasound images as the research object, and carries out the research with the idea of unsupervised domain adaptation.

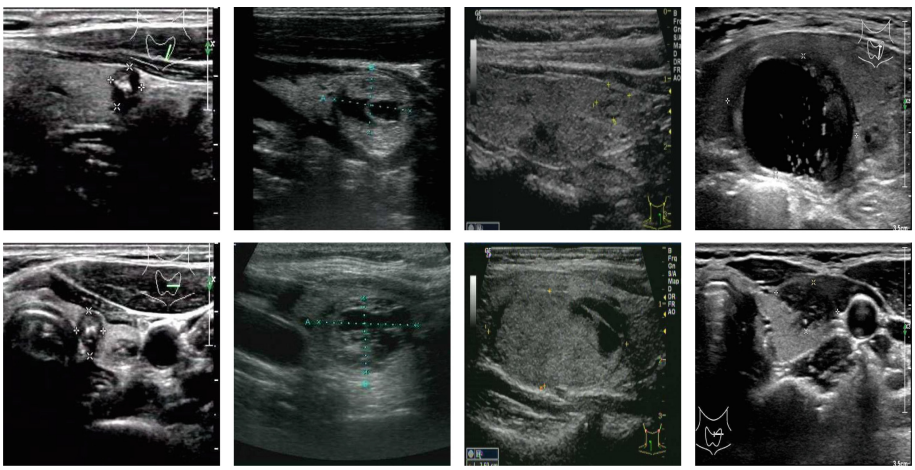



**Fig. 1.** Similarities and differences of visual features between ultrasonic images generated by different ultrasonic instruments.

Unsupervised domain adaptation research has achieved good results on benchmark natural datasets (e.g. mnist dataset [8], office-31 dataset [18]), which provides ideas for cross-domain research of medical images. However, the multi-model thyroid ultrasound image dataset used in this paper is different from the traditional natural dataset (e.g. the feature expression of the foreground area of the image is weak, and the size of the nodule tissue area is different). Such data characteristics bring difficulties to the feature extraction of domain adaptation networks. In addition, the adversarial domain adaptation methods inevitably lead to the feature offset of the target domain, so that the classification performance of the target domain in the testing phase is weakened to some extent. Although after domain adaptation the source domain samples are close to the target domain samples as a whole, there is still a distribution shift between the source domain and the target domain of the same category. It leads to the fact that the supervised source classifier shared by the two domains cannot accurately classify the target domain samples.

To solve the above problems, we propose a Multi-task Class feature space Fusion Domain Adaptation Network (MCFDAN). Firstly, the network combines adversarial learning with image reconstruction, which constructs a multi-task learning structure including adversarial learning, reconstruction, and classification. It ensures that the target domain samples will not be shifted during the fusion process. Secondly, to process ultrasound images containing nodules of different scales, we design a multi-dimensional adversarial structure for feature extraction. This structure introduces a recurrent cross-attention module [6] that can obtain global semantic information. Finally, we propose a Class feature Space Fusion (CSF) algorithm, which enables category alignment of source domain features and target domain features.

To sum up, the main innovations and contributions of this paper are summarized as follows:

- This paper proposes a Multi-task Class feature space Fusion Domain Adaptation Network, which enhances domain invariant feature extraction and mitigates feature offset of the target domain samples during encoding.
- This paper proposes a Class feature Space Fusion algorithm for classifier adaptation, which realizes the same category alignment of the source domain and the target domain.
- The MCFDAN achieves leading performance on the multi-model thyroid ultrasound medical imaging dataset and improves the generalizability of the model.

## 2 Related Work

### 2.1 Adversarial Domain Adaptation

Deep neural networks have strong advantages in image processing, so they are widely used in domain adaptation research. In addition, the generative adversarial strategy has become popular in the field of domain adaptation in recent

years. The idea of these methods are derived from Generative Adversarial Networks (GAN) [4]. Initially, Yaroslav et al. [2] proposed the Domain Adversarial Neural Network (DANN), which integrated the Gradient Reversal Layer (GRL) into the standard convolutional neural network architecture. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation (MCD-DA) [19] fixed the parameters of the feature extractor and the two classifiers respectively. The network performed a two-stage optimization to enhance feature expression so that the effects of the two classifiers are as the same as possible. The most recent method Adversarial Domain Adaptation with Domain Mixup (DM-ADA) [23] used soft scores to reasonably express images or features with unclear semantics. The above methods realized the feature space alignment by confusing the judgment of the domain discriminator on the sources of the input data, which inevitably lead to the feature shift of the target domain. To some extent, the classification performance of the target domain data in the testing phase is weakened.

There have also been many attempts by researchers to achieve classifier adaptation. Long et al. [12] added a residual architecture to the source classifier to alleviate the bias between the target sample and the source classifier, and achieved the category adaptation between the source and target tasks; Zhang et al. proposed the SymNet [26] to complete the learning of domain discrimination and confusion by designing additional classifiers with shared neurons. Most classifier adaptation methods rely on the design of complex network architectures, which bring high computational costs.

### 2.2 Medical Applications Based on Deep Learning

Artificial intelligence has made breakthroughs in medical-aided diagnosis. Kong et al. [13] first introduced convolutional neural networks into the detection of thyroid ultrasound images. Sun et al. [22] applied a Support Vector Machine (SVM) to classify features extracted by a CNN with an accuracy rate of 92.5%. Song et al. [20] proposed a Feature-enhanced Dual-branch Network (FDNet), which achieved competitive nodule detection performance. Yu et al. [25] proposed an edge self-attention erasure method for weakly supervised semantic segmentation. The network can obtain good nodule segmentation results with only image classification labels. Although the performance of the above methods has gradually reached the diagnostic result of professional doctors, most smart healthcare systems have poor generalization to multi-model medical images.

To improve the generalization ability of the model, domain adaptation is applied to medical aided diagnosis. In terms of medical image classification, Carolina et al. [14] proposed an unsupervised domain adaptation method to classify stem cell-derived cardiomyocytes. Ying et al. [24] designed a domain adaptation network for the classification task of thyroid ultrasound images. The network was used for domain adaptation between labeled and unlabeled thyroid ultrasound images in different domains. Zhao et al. [27] proposed a Semantically Consistent Generative Adversarial Network (SCGAN) for multi-modal medical image recognition. According to the characteristics of ultrasound images, the

feature extraction of the nodular region inside the image is a further research direction.

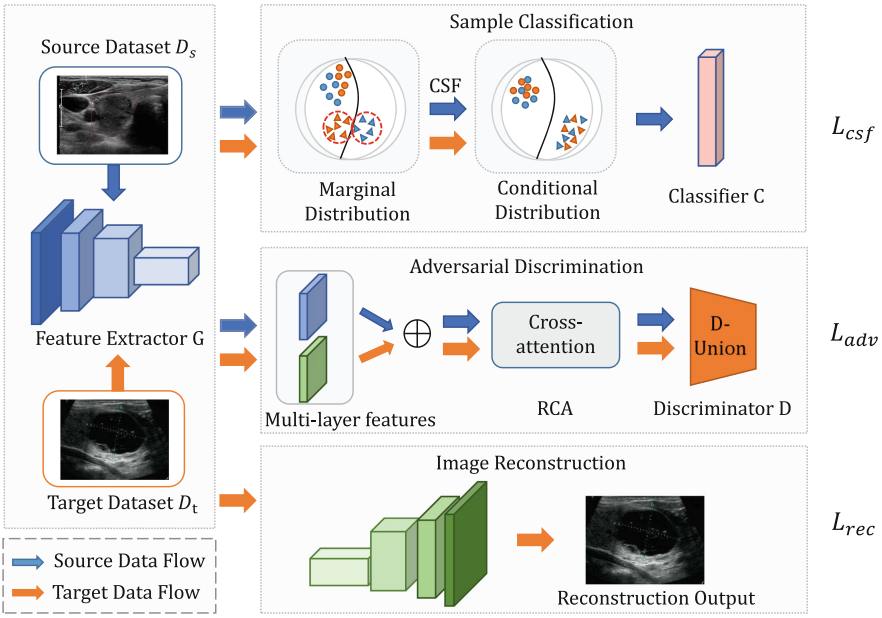



**Fig. 2.** The structure of the Multi-task Class feature space Fusion Domain Adaptation Network. In the sample classification task, the red circles indicate the source domain distribution and the target domain distribution of the same category, and there is an offset between them. The curve represents the classification decision boundary. In the adversarial discrimination task, we have a simplified representation for the fusion of multi-layer features, and D-Union denotes the union of D-down, D-mid, and D-up. (Color figure online)

## 3 Method

In the MCFDAN, we can obtain sufficiently labeled source domain data $D_s = \left\{\left(x_s^i, y_s^i\right)\right\}_{i=1}^{n_s}$ and unlabeled target domain data $D_t = \left\{\left(x_t^i\right)\right\}_{i=1}^{n_t}$, they are used as the training set, and the labeled target domain data $D_t = \left\{\left(x_t^i, y_t^i\right)\right\}_{i=1}^{n_t}$ is used as the test set. The source domain and the target domain are represented by the data distributions $P_s$ and $P_t$ respectively. To learn a shared domain space, we fuse the source domain with the target domain by aligning the data feature space.

The MCFDAN is mainly composed of four parts: feature extractor $G$, label classifier $C$, domain discriminator $D$, and recurrent cross-attention module $RCA$. The feature extractor $G$ is responsible for extracting the features of the data and

decoding the representation of the end layer feature embedding. The label classifier $C$ is mainly responsible for the classification of data features. The domain discriminator $D$ is also a classifier in essence, but the purpose of classification is to distinguish whether the feature comes from the target domain or the source domain. The recurrent cross-attention module (RCA) [6] is also an important part to enhance the correlation between pixels of different scale feature maps. Therefore, the overall optimization objective of the model is:

$$L_{total}(D_s, D_t) = L_{csf}(D_s) + \lambda_1 \cdot L_{adv}(D_s, D_t) + \lambda_2 \cdot L_{rec}(D_t) \tag{1}$$

Model label classification task $L_{csf}$, domain confrontation discrimination task $L_{adv}$ and target data reconstruction task $L_{rec}$ are jointly optimized. $\lambda_1$and $\lambda_2$ are the weights of adversarial learning and reconstruction learning. Next, we introduce the multi-task domain adaptation framework and class feature space fusion respectively.

### 3.1 Multi-task Domain Adaptation Framework

As shown in Fig. 2, the multi-task domain adaptation framework includes sample classification, image reconstruction, and adversarial discrimination. The feature extractor $G$ is the encoding branch. Specifically, the forward propagation includes four feature maps with different proportions, and the last output feature is adjacent to the classifier. The overall training process is carried out based on the marked source domain data samples, which is a supervised classification task.

The target domain image reconstruction is added as an auxiliary task to the adversarial domain adaptation method. The decoding branch is connected after the feature extractor $G$. We take the target domain feature map with strong semantic information as the input, and upsample it by linear interpolation to obtain high-resolution features. These different scale features are fused with the corresponding size features on the encoding branch. The mean square error (MSE) is used as reconstruction loss $L_{rec}$ to reduce the differences between the reconstructed image and the original input target image. The optimization objective and loss are as follows:

$$\min_{G} L_{rec} \tag{2}$$

$$L_{rec}(D_t) = \|x_t - fr(x_t)\|_2^2 \tag{3}$$

$x_t$ represents the input data of the original target domain, and $fr(x_t)$ represents the reconstructed data. In the process of domain feature space alignment, the $L_{rec}$ loss is reduced by reducing the Euclidean distance between $x_t$ and $fr(x_t)$, which mitigates feature offset of the target domain samples during the encoding process.

The shared feature extractor $G$ and the decoding branch are connected to construct a feature pyramid structure [9]. It has multi-layer semantic information, which can fully express pathological tissue regions of different sizes. The feature maps of three scales are fused in correspondence and passed to three

domain discriminators D-down, D-mid, and D-up respectively. It is worth noting that the model introduces the RCA module [6] into each adversarial structure, which can extract the global semantic information of pathological tissue and promote domain invariant feature extraction. The function $f_k(x)$ represents the discrimination of domains:

$$f_k(x) = D_k(f_{rca}(G_k(x)))(k = down, mid, up) \tag{4}$$

$f_{rca}$ denotes the recurrent cross-attention attention calculation process. The minimum and maximum games are performed on the multi-scale features to achieve the confusion of source and target domain features. Three independent discrimination losses are defined on the domain and the target domain: $L_{adv}^{down}$, $L_{adv}^{mid}$, and $L_{adv}^{up}$. The optimization objective and loss are as follows:

$$\min_{G_{down}} \max_{D_{down}} L_{adv}^{down}, \min_{G_{mid}} \max_{D_{mid}} L_{adv}^{mid}, \min_{G_{up}} \max_{D_{up}} L_{adv}^{up} \tag{5}$$

$$\begin{aligned} &L_{adv}(D_s, D_t) = L_{adv}^{down} + L_{adv}^{mid} + L_{adv}^{up} \\ &= E_{(x_s,y_s)\sim P_s(x_t)\sim P_t}[-\log([f(x_s^i), f(x_t^j)])] \\ &= -\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}[d_s^i, d_t^j]\log([f_{down}(x_s^i), f_{down}(x_t^j)]) \\ &\quad -\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}[d_s^i, d_t^j]\log([f_{mid}(x_s^i), f_{mid}(x_t^j)]) \\ &\quad -\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}[d_s^i, d_t^j]\log([f_{up}(x_s^i), f_{up}(x_t^j)]) \end{aligned} \tag{6}$$

Among them, $\left[f_k(x_s^i), f_k(x_t^j)\right]$ is the cascade of domain prediction, $[d_s, d_t]$ is the cascade of domain labels, $n_s$ and $n_t$ are the number of samples in the source domain and the target domain respectively. The above training process make the distribution of features in the source and target domains converge, which is an adversarial discrimination task.

In summary, we construct a multi-task domain adaptation network, including sample classification, image reconstruction, and adversarial discrimination. The network not only obtains multi-dimensional task-dependent deep domain invariant features, but also mitigates feature offset of the target domain samples.

### 3.2 Class Feature Space Fusion

In the above multi-task domain adaptation framework, the network optimized by adversarial discrimination and image reconstruction has fused the source domain feature space and target domain feature space as a whole. The output source domain features are passed into the classifier for judgment. The classifier $C$

Initially uses cross-entropy loss $L_{cls}$ to optimize, and the optimization objective and loss are as follows:

$$\min_{G,C} L_{cls} \tag{7}$$

$$\begin{aligned} L_{cls}(D_s) &= -E_{(x_s,y_s)\sim P_s}\log(C(G(x_s^i))) \\ &= -\frac{1}{n_s}\sum_{i=1}^{n_s} y_s^i \log(C(G(x_s^i))) \end{aligned} \tag{8}$$

In the above equation, $G\left(x_s^i\right)$ represents the feature extracted by the model, $y_s^i$ is the true class label, and $E_{(x_s,y_s)\sim P_s}$ represents the expectation of conforming to the source domain class feature spatial distribution $P_s$. As the sample size increases, the arithmetic mean of the sample will approach the expectation of the source domain feature space, and the loss is minimized to train the classification model.

However, As shown in Fig. 2, the marginal distribution of the same class is still different between the source samples and target domain samples, which directly leads to the poor adaptation of the supervised source classifier to the target samples. In the sample classification task, the class feature space fusion algorithm is introduced to fuse the category space of the source domain and the target domain after the feature extractor. Specifically, for each category $i$, $\mu_{si}$ and $\mu_{ti}$ are used to represent the average feature vectors of the source and target domains, respectively. The average inter-domain difference $\Delta\mu_i = \mu_{ti} - \mu_{si}$ is the source domain feature migration direction, which can mitigate the semantic offset of the two domains as a whole.

The average inter-domain difference is not sufficient to determine the direction of source domain sample feature migration, and we introduce standard deviation for the target samples of each category to expand the intra-class semantic enhancement (e.g., the brightness of calcified points):

$$\sigma_t^j = \sqrt{\frac{\sum(G(x_t^j) - \mu_t^j)}{N}} \tag{9}$$

$G(x_t^j)$ and $\mu_t^j$ are the depth space feature vector and mean vector of the migrated target samples respectively. In summary, we can obtain the joint distribution $N = (\Delta\mu_i, \sigma_t^i)$ of $i$ categories. The source domain features $G(x_s^i)$ of each category can be subjected to various semantic transformations along the random sampling direction of $N = (\alpha\Delta\mu_i, \alpha\sigma_t^i)$. The enhanced features $G'(x_s^i) \sim N(G(x_s^i) + \alpha\Delta\mu_i, \alpha\sigma_t^i)$ that fit the target domain category feature space distribution are obtained by the above transformations, where $\alpha$ is the parameter that controls the enhancement intensity. Here $\alpha$ takes the value of $t_0/t$, where $t_0$ and $t$ are the number of current iteration round and the maximum iteration round respectively. As the number of training rounds increased, the intensity of the standard deviation enhancement gradually increased.

The original cross-entropy classification loss $L_{cls}$ is updated by the augmented source features and the corresponding supervised labels. The method

further achieves conditional distribution alignment on the basis of ensuring uniform edge distribution. The optimization objective and loss are as follows:

$$\min_{G,C} L_{csf} \tag{10}$$

$$\begin{aligned} L_{csf}(D_s) &= -E_{(x_s,y_s)\sim P_s}\log(C(G'(x_s^i))) \\ &= -\frac{1}{n_s}\sum_{i=1}^{n_s} y_s^i \log(C(G'(x_s^i))) \end{aligned} \tag{11}$$

## 4 Experimental Results and Analysis

### 4.1 Dataset

**Medical Thyroid Ultrasound Dataset.** This study has been approved by the Medical Ethics Committee of Tianjin Medical University Cancer Institute and Hospital. Written consent had been obtained from each patient after full explanation of the purpose and nature of all procedures used. The thyroid ultrasound image data in this paper is collected from Tianjin Medical University Cancer Institute and Hospital. These images come from patients of all ages, and each image contains at least one nodule. The dataset is divided into four different visual style domains: represented by F, T, U, and P respectively according to the type and setting of the ultrasound instrument. All images have credible benign or malignant labels annotated by professional radiologists. The distribution of the thyroid ultrasound dataset is shown in Table 1.

**Table 1.** Constitution of the medical thyroid ultrasound dataset

| – | Benign | Malignant | Domain samples |
|---|---|---|---|
| P | 1995 | 2000 | 3995 |
| T | 585 | 585 | 1170 |
| U | 764 | 764 | 1528 |
| F | 101 | 55 | 156 |
| Class samples | 3445 | 3404 | 6849 |

### 4.2 Experiment Setup and Evaluation Value

For a fair comparison, we use ResNet50 pre-trained on ImageNet [17] as the backbone network. In this paper, experiments are conducted based on Intel Core i7-7700k processor and single NVIDIA TITAN RTX graphics card. The deep learning framework is PyTorch 1.1 and the development language version is python 3.7. In the process of training the domain adaptation network, to achieve

fast convergence of the model, we use the stochastic gradient descent (SGD) algorithm as an optimization strategy. The learning rate is set to 0.001 and the momentum is set to 0.9, and the batch size is set to 32. During the test process, the batch size is set to 4, and the weight $\lambda_1$ and weight $\lambda_2$ are both set to 1. All images are preprocessed. The image size of the medical thyroid ultrasound dataset is adjusted to $224 \times 224$ pixels. The number of model training iterations is set to 50000, and the model is simultaneously fitted to the training dataset.

In order to objectively evaluate the performance of the smart healthcare model proposed in this paper, the evaluation value is the image classification accuracy of the model in the target domain. The calculation is as follows:

$$Acc(D_t) = E_{x_t \in D_t}(y_t^i = C(G'(x_t^i))) \tag{12}$$

In the above equation, $y_t^i$ represents the real class label of image $x_t$ in the target domain. The test accuracy of the target domain data is the only evaluation value to measure the performance of the model.

### 4.3 Experiment Results

To show the performance of MCFDAN for adaptive tasks in the medical field, we compare it with various mainstream unsupervised adaptive methods. In order to verify the effect of the unsupervised adaptive method, we performed domain adaptation based on 9 directions on four domains: $P \rightarrow T$, $P \rightarrow U$, $P \rightarrow F$, $T \rightarrow P$, $T \rightarrow U$, $T \rightarrow F$, $U \rightarrow P$, $U \rightarrow T$, $U \rightarrow F$. Here the F dataset is not set as the source domain, because the number of ultrasound images in the F dataset is small. The model without constraints may lead to overfitting of the training data.

**Table 2.** Classification accuracy (%) of target domain on the medical thyroid ultrasound dataset. The best performance is indicated in **bold**.

| Method | $P \rightarrow T$ | $P \rightarrow U$ | $P \rightarrow F$ | $T \rightarrow P$ | $T \rightarrow U$ | $T \rightarrow F$ | $U \rightarrow P$ | $U \rightarrow T$ | $U \rightarrow F$ | Average |
|---|---|---|---|---|---|---|---|---|---|---|
| ResNet (Source Only) [5] | 65.128 | 78.469 | 69.231 | 67.159 | 70.924 | 49.359 | 75.995 | 67.521 | 83.333 | 69.680 |
| DAN [10] | 79.658 | 88.481 | 83.333 | 70.638 | 72.971 | 71.153 | 81.076 | 71.282 | 84.615 | 78.134 |
| RevGrad [1] | 81.197 | 87.434 | 86.538 | 68.185 | 72.055 | 73.076 | 80.401 | 70.427 | 82.692 | 78.001 |
| D-CORAL [21] | 80.598 | 88.219 | 82.692 | 69.862 | 71.157 | 73.076 | 78.798 | 70.512 | 85.256 | 77.797 |
| DANN [2] | 87.436 | 85.668 | 88.462 | 79.975 | 79.188 | 85.897 | 84.406 | 75.812 | 87.179 | 83.780 |
| CDAN [11] | 88.547 | 87.500 | 90.385 | 82.854 | 83.966 | 86.538 | 85.707 | 74.017 | 90.380 | 85.544 |
| MRAN [28] | 85.982 | 91.557 | 87.179 | 80.976 | 81.086 | 81.410 | 86.983 | 75.726 | 84.615 | 83.946 |
| MSDAN [24] | **91.667** | 94.241 | 94.231 | 89.237 | 88.154 | 89.744 | 87.409 | **84.274** | 91.667 | 90.069 |
| Our method(w/o CSF) | 90.342 | 93.848 | 95.513 | 89.186 | 88.220 | 91.667 | 91.589 | 84.103 | 90.385 | 90.539 |
| Our method(w CSF) | 90.865 | **94.568** | **96.154** | **91.940** | **88.257** | **93.590** | **93.442** | 83.564 | **95.513** | **91.988** |

**Table 3.** Effectiveness of Domain Adaptation (DA), Recurrent Cross-Attention Module (RCA), and Class feature Space Fusion (CSF) algorithm on the medical thyroid ultrasound dataset. The best performance is indicated in **bold**.

| Variants | DA | RCA | CSF | $P \rightarrow T(\%)$ | $\triangle(\%)$ |
|---|---|---|---|---|---|
| ResNet50 | – | – | – | 65.128 | – |
| Baseline | √ | – | – | 87.436 | +22.308 |
| Baseline + RCA | √ | √ | – | 89.573 | +24.445 |
| Baseline + CSF | √ | – | √ | 90.231 | +25.103 |
| MCFDAN | √ | √ | √ | **90.865** | **+25.737** |

Table 2 shows the comparison of the experimental results between our method and other mainstream methods on the thyroid ultrasound dataset. The result reveals several interesting observations: (1) The ResNet (source only) method in the table represents feature extractors and classifiers without domain adaptation. The proposed method achieves the most advanced performance in all domain adaptation directions. In general, the average accuracy of all transfer tasks increases from 69.680% to 90.539%. (2) The CSF improves the average classification accuracy by approximately 1.4%, which is a significant enhancement to the overall adaptability of the model. The above results show that the MCFDAN has achieved excellent performance in multi-model thyroid ultrasound data, which is meaningful for improving the generalization of smart healthcare systems.

Table 3 shows the classification results of the ablation study. The DANN method is set as the baseline network for ablation studies. The experimental results show that domain adaptation has better results in terms of sample classification compared with ResNet50. Besides, after adding the RCA module and the CSF algorithm, our experimental results show strong competitiveness in average accuracy. In summary, each module introduced in this article has a positive impact on the domain adaptation process.

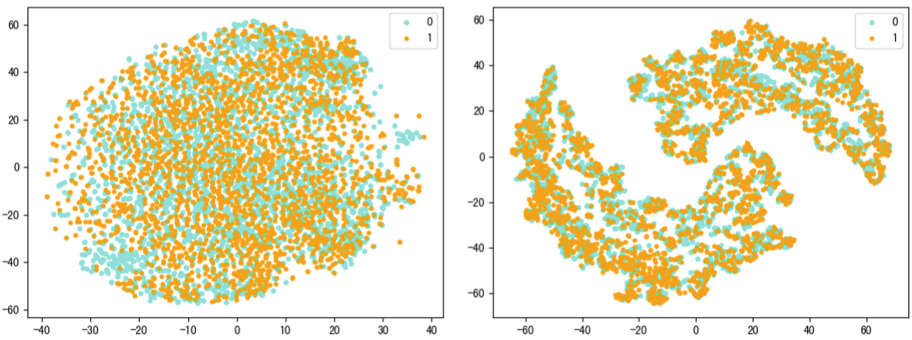



**Fig. 3.** t-SNE Visualization of sample distribution of Philips→Toshiba task.

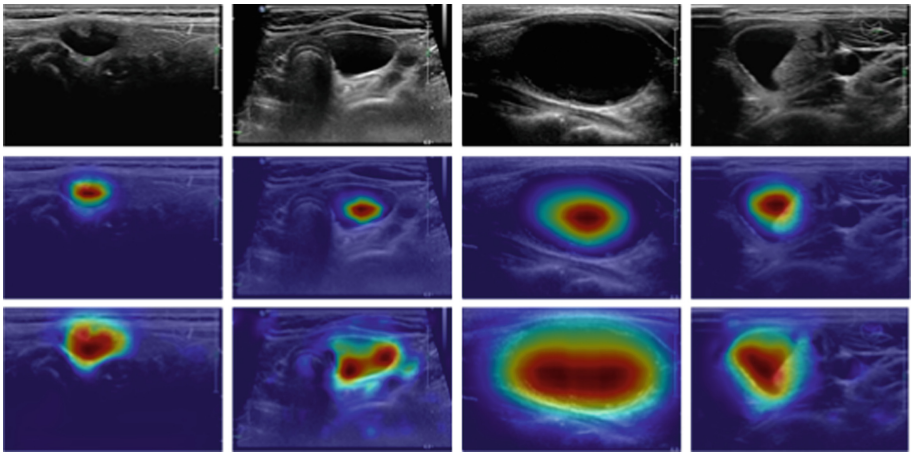


**Fig. 4.** Visualization of RCA module optimization. The first row in the figure represents the input images, the second row and the third row represent the visualization effect before and after adding the RCA module, respectively.

Visualization experiments are conducted to verify model validity. Figure 3 shows the t-SNE visualization of the sample distribution on the thyroid ultrasound image dataset. "0" and "1" in the figure are the labels of the two domains respectively. In the left figure, there are statistical distribution differences between the domains before adaptation. In the right figure, the MCFDAN proposed in this paper can effectively reduce the distribution offset between the image embedded features in the source domain and the target domain, and realize the alignment of the same class images between domains. Figure 4 shows the RCA activation effect. After adding this module, the activation area is effectively extended to the global area of the nodular tissue, and the effect of feature extraction is more obvious.

## 5 Conclusion

Smart healthcare systems have greatly improved diagnosis efficiency and accuracy. However, the current deep networks have poor generalization for medical multi-models data. To solve the problem of model performance degradation caused by domain offset, this paper constructs a Multi-task Class Feature Space Fusion Domain Adaptation Network. On the one hand, the multi-task domain adaptation framework mitigate feature offset of the target domain and effectively enhances the domain invariant feature extraction. On the other hand, the class feature space fusion algorithm improves the generalizability of the source domain classifier. The experimental results show that the MCFDAN has achieved excellent performance on the multi-model medical thyroid ultrasound dataset, which improves the generalization of smart healthcare systems.

**Acknowledgments.** This work was supported by Major Scientific and Technological Projects for A New Generation of Artificial Intelligence of Tianjin (Grant No. 18ZXZNSY00300).

## References

1. Ganin, Y., Lempitsky, V.S.: Unsupervised domain adaptation by backpropagation. In: Proceedings of the 32nd International Conference on Machine Learning, vol. 37, pp. 1180–1189. JMLR.org, Lille, France (2015)
2. Ganin, Y., et al.: Domain-adversarial training of neural networks. In: Csurka, G. (ed.) Domain Adaptation in Computer Vision Applications. ACVPR, pp. 189–209. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58347-1_10
3. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587. IEEE Computer Society, Columbus, OH, USA (2014)
4. Goodfellow, I.J., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. IEEE Computer Society, Las Vegas, NV, USA (2016)
6. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: CCNet: criss-cross attention for semantic segmentation. In: International Conference on Computer Vision, pp. 603–612. IEEE, Seoul, Korea (South) (2019)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1106–1114, Lake Tahoe, Nevada, United States (2012)
8. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
9. Lin, T., Dollár, P., Girshick, R.B., He, K.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 936–944. IEEE Computer Society, Honolulu, HI, USA (2017)
10. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: Bach, F.R., Blei, D.M. (eds.) Proceedings of the 32nd International Conference on Machine Learning, vol. 37, pp. 97–105. JMLR.org, Lille, France (2015)
11. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems, pp. 1647–1657. Montréal, Canada (2018)
12. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. In: Advances in Neural Information Processing Systems, pp. 136–144. Barcelona, Spain (2016)
13. Ma, J., Wu, F., Jiang, T., Zhu, J., Kong, D.: Cascade convolutional neural networks for automatic detection of thyroid nodules in ultrasound images. Med. Phys. **44**(5), 1678–1691 (2017)
14. Pacheco, C., Vidal, R.: An unsupervised domain adaptation approach to classification of stem cell-derived cardiomyocytes. In: Shen, D., Liu, T., Peters, T.M., Staib, L.H., Essert, C., Zhou, S., Yap, P.-T., Khan, A. (eds.) MICCAI 2019. LNCS, vol. 11764, pp. 806–814. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32239-7_89

15. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010)
16. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99. Montreal, Quebec, Canada (2015)
17. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
18. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_16
19. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3723–3732. IEEE Computer Society, Salt Lake City, UT, USA (2018)
20. Song, R., et al.: Dual-branch network via pseudo-label training for thyroid nodule detection in ultrasound image. Appl. Intell. **52**, 11738–11754 (2022)
21. Sun, B., Saenko, K.: Deep CORAL: correlation alignment for deep domain adaptation. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 443–450. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_35
22. Sun, C., et al.: Evaluation of a deep learning-based computer-aided diagnosis system for distinguishing benign from malignant thyroid nodules in ultrasound images. Med. Phys. **47**(9), 3952–3960 (2020)
23. Xu, M., Zhang, J., Ni, B.: Adversarial domain adaptation with domain mixup. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 6502–6509. AAAI Press, New York, NY, USA (2020)
24. Ying, X., et al.: MSDAN: multi-scale self-attention unsupervised domain adaptation network for thyroid ultrasound images. In: Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, pp. 871–876. IEEE, Virtual Event, South Korea (2020)
25. Yu, M., Han, M., Li, X., Jiang, H., Chen, H., Yu, R.: Adaptive soft erasure with edge self-attention for weakly supervised semantic segmentation: thyroid ultrasound image case study. Comput. Biol. Med. **144**, 1–17 (2022)
26. Zhang, Y., Tang, H., Jia, K., Tan, M.: Domain-symmetric networks for adversarial domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5031–5040 (2019)
27. Zhao, J., et al.: Semantic consistency generative adversarial network for cross-modality domain adaptation in ultrasound thyroid nodule classification. Appl. Intell. **52**(9), 10369–10383 (2022)
28. Zhu, Y., Zhuang, F., Wang, J.: Multi-representation adaptation network for cross-domain image classification. Neural Netw. **119**, 214–221 (2019)

# A Caching Strategy Based on Spreading Influence in Information-Centric Satellite Networks

Haowei Wang, Rui Xu, Xiaoqiang Di(✉), Jing Chen, Dejun Zhu, Juping Sun, and Yuchen Zhu

School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
dixiaoqiang@cust.edu.cn

**Abstract.** Low-Earth-Orbit (LEO) satellite networks can transmit data globally, but their time-varying nature makes data transmission delayed and inefficient. An important feature of ICN (Information-Centric Network) is in-network caching, which is helpful to increase transmission efficiency. To reduce the time-varying effect on data transmission and to improve transmission efficiency among satellites, this paper introduces ICN into LEO satellite networks and proposes a Caching Strategy based on Spreading Influence (CSSI) for satellite networks. Our strategy portrays satellite nodes and popular contents in terms of spreading influence, and caches popular contents with dynamic probability in satellite nodes with high global spreading influence in the period to obtain a better caching effect. Compared with LCE, LCD, Prob, and Betw strategies, the CSSI effectively improves the cache hit ratio and reduces the average content request delay.

**Keywords:** Information-centric satellite networks · Spreading influence · Cache nodes · Popularity

## 1 Introduction

With the advancement of technology and the advent of globalization, people's demand for fast and stable global communication is gradually obvious. For traditional terrestrial networks, the communication effect is easily affected by geographical location, some inaccessible areas have poor communication effects or even no communication at all due to the inability to deploy terrestrial base stations. Unlike terrestrial networks, satellite networks cover a wide area and don't rely too much on the construction of terrestrial base stations, they have great advantages in global communication, disaster response, reliability, etc., these advantages cause the country and the industry to increase investment

This research was funded by the National Natural Science Foundation of China under grant No. U21A20451, the Science and Technology Planning Project of Jilin Province, China (20200401105GX), and the China University Industry-Academia-Research lnnovation Fund un-der grant No. 2021FNA01003.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 153–164, 2022.
https://doi.org/10.1007/978-3-031-19208-1_13

in the research and development of satellite communication technologies [1]. Among them, LEO satellites have attracted extensive attention because of their advantages of seamless global coverage, short transmission delay, and low launch cost.

However, LEO satellites in different orbits have short contact time and low data transmission efficiency. ICN identifies contents with uniform and consistent naming rules and can retrieve data based on any location in the network by naming it and the data is shared across the network, this model has better mobile applicability [2]. One of the most important features of ICN is its ability to provide transparent and ubiquitous caching across the network, which greatly reduces the distance and time for users to obtain data. Using the in-network caching technology of ICN in satellite networks can effectively increase transmission performance.

The traditional ICN caching strategies don't apply to satellite networks with mobility, when satellite nodes move at a high speed, it is often accompanied by the on-off of inter-satellite links and the change of transmission paths, traditional caching strategies will be slow to match or even fail to get a match for cached data, which greatly reduces the effectiveness of the caching strategy [3]. Based on this, this paper proposes a caching strategy applicable to satellite networks with the following contributions.

- By using the concept of complex networks, the k-shell method together with information entropy is used to represent the spreading influence of satellite nodes. The k-shell method can efficiently and accurately identify satellite nodes with high spreading influence in the network, while information entropy can classify the spreading influence at a finer granularity from the perspective of neighboring nodes.
- The characteristic of satellite networks' time slots duration and the spreading influence within a single time slot are synthesized to obtain the global spreading influence of satellite nodes, and a caching algorithm based on the global spreading influence and content popularity is designed to cache popular contents with dynamic probability in satellite nodes with high global spreading influence, and the effectiveness of this strategy is demonstrated by using ndnSIM as the experimental environment.

## 2 Related Work

The current caching strategies in ICN can be divided into collaborative caching strategies and non-collaborative caching strategies [4]. Non-collaborative caching strategies mean that the caching decision is made without considering the overall distribution of the network and the nodes make caching decisions independently. The LCE [5] (Leave Copy Everywhere) is a non-collaborative caching strategy, the return data is cached indistinguishably in each passing node. Collaborative caching is further divided into explicit collaborative caching and implicit collaborative caching. Explicit collaborative caching sets up a global controller that processes global information for caching decisions, which can effectively use caching resources and improve caching efficiency, with the disadvantage that it requires too much computing power. Implicit collaborative caching makes caching decisions based on some additional information (such as the importance of nodes, content popularity, caching probability, etc.), where LCD [6] (Leave Copy Down), MCD [6] (Move Copy Down), Prob [6] (Copy with Probability), ProbCache [6] are all implicit collaborative caching strategies.

In addition to the above strategies, researchers have achieved many results in other aspects of implicit collaborative caching strategies. Chai et al. [7] calculated the betweenness centrality of each node in the network topology and cached the data in the nodes with large betweenness centrality, which improved the cache hit ratio and reduced the transmission delay compared to the above strategies. Cai et al. [8] calculated the degree centrality, closeness centrality, and betweenness centrality of each node in the network topology, and used them as indicators to comprehensively evaluate the importance of each node with the cache space availability rate to select nodes for content caching, which effectively improved the cache hit ratio. Li et al. [9] used content centrality as an indicator to cache popular contents with a higher probability in topologically centered nodes, which greatly reduced cache redundancy and improved cache utilization. Man et al. [10] noticed the problem of correlation between cached contents, by considering the correlation between cached content and other contents in the cache nodes, the data is cached in the nodes with a higher correlation with it, which effectively improved the cache hit ratio. The effectiveness of the above caching strategies has been proven in static networks, however, satellite networks are time-varying and periodic, and the transmission paths of interest packets and data packets change with the topology.

To improve the applicability of ICN caching strategies in satellite networks, the first problem to be solved is the time-varying problem. Zhu et al. [11] dealt with satellite networks in a virtual topology approach, evaluated satellite nodes' importance in terms of node betweenness, node closeness, and node distance within each static time slot, and obtained the integrated importance of satellite nodes over the entire operation period by using the time slot length as a weight. Xu et al. [12] established a three-layer satellite network model to calculate the popularity of each content, and the caching probability is proportional to the popularity, this strategy is compared with the traditional IP strategy, no-cache strategy, and LCE strategy, and it achieved a better result in terms of average content access delay. Salvatore et al. [13] demonstrated the effectiveness of content popularity in caching by caching contents of common interest to different users on satellites. The above strategies prove the feasibility of on-board ICN caching, but when considering the ability of satellite nodes to spread data, only the spreading effect of satellite nodes themselves is considered and the difficulty of placing cache due to the difference in spreading effect of satellite nodes in different time slots is not considered.

In this paper, the satellite networks' period is sliced in seconds to obtain a topological set, and a series of steady-state topological time slots are obtained by merging time slices with the same topology. For the problem that the current strategies only consider the spreading influence of satellite nodes themselves, this paper proposes to use the spreading influence of neighboring nodes to participate in evaluating the comprehensive spreading influence. For the problem that the spreading influence of satellite nodes varies greatly in different time slots, this paper proposes to analyze the global spreading influence of the satellite nodes by combining the time slot length, which is helpful to find the applicable cache nodes to place the popular contents.

## 3 The Time-Varying Model of Satellite Networks

As the medium of data exchange between satellites, only stable inter-satellite links can ensure efficient transmission [14], however, the inter-satellite links of satellite networks

have large on-off variability in the period, and the topology of satellite networks will constantly change with the inter-satellite links, therefore, this paper needs to obtain a stable and limited number of topologies according to the periodic change characteristics of satellite networks.

As predictable time-varying networks, satellite networks can be viewed as different topological snapshots connected throughout their period. The location of satellite nodes and inter-satellite links change with time, but the total number of satellite nodes doesn't change. In this paper, the period T of satellite networks is sliced in seconds to obtain N time slices, and the resulting ensemble of time slices is denoted by $S = (S_1, S_2,\ldots, S_n)$, as shown in Fig. 1.

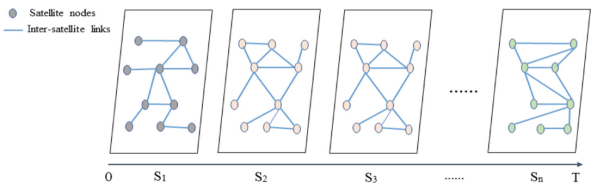



**Fig. 1.** Time slices model of satellite networks.

In this paper, the topology within each time slice of the satellite networks is expressed as an undirected connectivity graph, and the connectivity of satellite nodes is expressed in the form of an adjacency matrix as G = (V, E), where V denotes the number of satellite nodes in satellite networks and E denotes the inter-satellite links. In the adjacency matrix, a 0 indicates that there is no inter-satellite link between satellite nodes, a 1 indicates that there is an inter-satellite link between satellite nodes, and in particular, a 0 between the nodes themselves. As shown in Fig. 2.

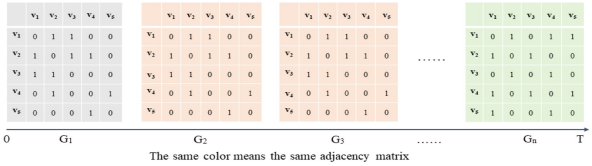



**Fig. 2.** Adjacency matrix graphs of satellite networks time slices.

In the time slice set S, the number of satellite nodes and inter-satellite links in every time slice $S_i$ doesn't change anymore. During the period of satellite networks, the inter-satellite links are not on and off all the time, and the inter-satellite links don't change in some fixed slots, so there are many time slices with the same topology in the time slice set S. The topologically identical time slices are merged to obtain a series of steady-state time slots, and in this paper, the merged time slots are denoted by $S_{new} = (G, T)$, where G denotes the topology of time slots after merging corresponding to the form of adjacency matrix, denoted by $G = (G_1, G_2,\ldots, G_n)$, T denotes the duration of each time slot, denoted by $T = (t_1, t_2,\ldots, t_n)$, $G_i$ denotes the topology of adjacency matrix form

under the corresponding period $t_i$, and $t_i$ denotes the duration of the corresponding stable topology i. The merging process is shown in Fig. 3.

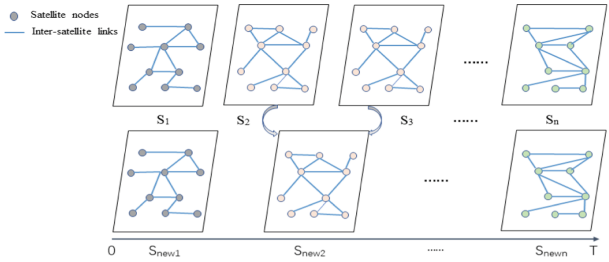



**Fig. 3.** Time slices merging process for the same topology of satellite networks.

By merging time slices of the same topology, the number of time slices can be greatly reduced. In the obtained set of time slots, the duration of some time slots is much larger than others, and satellite nodes can transmit more data within these time slots, so the importance of these time slots is considered relatively high in this paper. In the whole period T of satellite networks, the time slots with longer duration tend to be of higher research value, therefore, the weight of the duration of each time slot for the whole period is used as the weight to obtain the relative importance set W of each time slot, and the formula is as follows.

$$W = [w_1, w_2, \cdots, w_n] = \left[t_1/T, t_2/T, \cdots, t_n/T\right] \tag{1}$$

## 4 The CSSI Cache Model

The spreading influence of each satellite node is variational at different times, for the period, a satellite node with great spreading influence at a certain time slot doesn't necessarily have great influence in the period, so finding the nodes with great spreading influence in the whole period to deploy cache and improve data transmission efficiency while saving network resources is one of the purposes in this paper.

### 4.1 Cache Strategy Based on Spreading Influence

**Cache Nodes Selection Strategy** Based on Sect. 3, the time-varying satellite networks are transformed into a set of stable topological networks. With the help of complex network theory, some nodes are more important than others, and these nodes also play a greater role in information transmission [15]. Quickly and accurately identify the most influential spreading sources in the satellite networks, which plays an important role in effectively using limited resources and controlling the information spreading.

The k-shell method can effectively identify the most influential nodes in complex networks with a low time complexity, which is more suitable for satellite nodes with limited computing power, its core idea is to recursively remove all nodes in the network

whose degree value is less than or equal to k, at the end of recursion, all removed nodes form the kth kernel, thus dividing the network into a hierarchy from the core to the edge. The process of the k-shell method is as follows, the network topology is represented by an undirected connected graph G = (V, E). Firstly, the nodes with a moderate value of 1 are stripped, and the edge E connected to them is also removed, if there are still nodes with degree 1, repeat the preceding operations until no nodes with degree 1, then the nodes with degree 2 are removed, and the same process as above, finally the set of nodes with degree ks is obtained until all the nodes in the network are removed and the set of nodes with degree 1 to ks is obtained.

However, the k-shell method can only get node sets of different importance levels. Nodes in the same level can't be further divided, so it is impossible to better judge which nodes exert greater influence in data transmission. The importance of nodes is not only related to their attributes but also can be represented by the attributes of their neighbors. When the neighboring nodes of a node are more important, the node is more helpful for data transmission. In this paper, node information entropy is introduced to participate in calculating node spreading influence. Node information entropy takes into account the spreading effect of neighboring nodes. The greater the information entropy of a node, the greater its influence on the node. For the steady-state satellite networks topology graph G = (V, E), the following equation is available.

The importance of the node $I_i$ is expressed as the weight of this node degree value k to the sum of all satellite nodes' degree values.

$$I_i = \frac{k_i}{\sum_{j=1}^{N} k_j} \tag{2}$$

where $k_i$ denotes the degree value of satellite node i and N is the number of nodes in the satellite network topology graph G.

The information entropy of satellite nodes refers to the neighboring nodes' spreading effect in this paper, and the greater the information entropy of a node, the greater the influence.

$$e_i = -\sum_{j \in \delta(i)} I_j \times \ln I_j \tag{3}$$

where δ(i) is the neighborhood node set of satellite node i.

The node spreading influence in the satellite network topology graph G = (V, E).

$$p = \alpha ks + \beta e \tag{4}$$

where ks denotes the ks value of each satellite node, e denotes the information entropy value of the node, both of them are normalized, and α and β denote the weights of the two indicators, both of them are taken as 0.5 in this paper.

The global spreading influence of satellite nodes is represented by the time slot weight w and the spreading influence p within the corresponding time slot.

$$P = w_1 p_1 + w_2 p_2 + \cdots + w_n p_n \tag{5}$$

where $w_i$ is the weight of corresponding time slot i in the satellite networks' period.

So far, our strategy has obtained the global spreading influence ranking of satellite nodes, when a certain percentage of satellite nodes with high influence ranking were selected for the experiment, it was found that simply changing the number of satellite cache nodes didn't change the caching effect significantly over the period. Without degrading the caching effect while saving network resources as much as possible, the top 10% of satellite nodes in terms of global spreading influence are selected as caching nodes to verify the caching effect of this scheme in satellite networks.

**Contents Caching Probability Strategy**

Popular contents have more requests and will get wider spreading in the network. In this paper, the number of requests for each content in each user node is used to represent the content spreading influence, and the caching probability of content i is dynamically calculated from the users' perspective. The more requests for content, the greater content spreading influence and the greater caching probability in the cache nodes. When a data request node generates an interest packet, the corresponding name tag and the number of requests are recorded. The caching probability of content i in cache nodes for each data request node is as follows.

$$L_i = \frac{c_i}{\sum_{j=1}^{M} c_j} \tag{6}$$

where $c_i$ denotes the number of requests for content i in each data request node and M denotes the number of contents in each data request node.

### 4.2 The CSSI Routing and Forwarding Strategy

In our strategy, each data request node maintains a modified PIT [16] (Pending Interest Table) table. The table records the data tags, forwarding ports, number of generations, and cache probability. When the data request node requests data, it first checks its PIT, if the entry already exists, it updates the number of requests, while calculating and updating the cache probability based on Eq. 6; if the entry doesn't exist, it creates the corresponding entry, updates the number of requests and the cache probability, then generates an interest packet with the name tag, cache probability, and the cache node number. In the PIT of other satellite nodes, the data tags and forwarding ports are recorded, when a node has already received an interest packet, the new incoming port is added to the entry and the interest packet is discarded, otherwise, a new entry is created.

**Interest Packets Processing:** When the data request node forwards an interest packet according to the FIB [16] (Forwarding Information Base), the forwarding port is recorded in the entry. When the interest packet arrives at the next satellite node, the satellite node will query the interest packet for number matching, if the match is successful, it will check whether its CS [16] (Content Store) contains the corresponding data packet, and if the packet exists in the CS, it will modify the cache probability according to the interest packet and return the packet directly to the incoming port, if the packet doesn't exist in the CS, the satellite node will query whether the interest packet has been received in

PIT, if yes it will record the incoming port under the corresponding entry, if no it will create the PIT entry. The number matching failure will directly query the PIT. For ICN, an interest packet can only retrieve one data packet, thus achieving traffic balance. The satellite node then forwards the interest packet to the next satellite node based on the FIB so that it gradually approaches the data production node or the satellite node that has the content.

**Data Packets Processing:** When the data production node receives an interest packet, it generates the data packet with the corresponding cache probability and cache satellite node number, if it hits the cache nodes, the cache probability in the data packet will be modified according to the interest packet. When the data packet is delivered to the next satellite node, according to our strategy, if the satellite node performs the data caching function, it first queries its CS, and if there is no data packet, it caches it with the corresponding probability, then queries its PIT entry, if there is a record, it deletes the entry based on all incoming ports outgoing packets in the entry, and if the satellite node only performs the forwarding function, it directly queries its PIT for forwarding until it returns the packet to the data request node.

## 5 Simulation Experiments

### 5.1 Parameter Settings

The experiments in this paper were conducted on the network simulation platforms NS-3 [17] and ndnSIM [17]. This paper uses the Iridium system [18] as the experimental object, the Iridium system with 66 satellite nodes distributed in six polar planes to achieve seamless coverage of the Earth, which is one of the operational satellite constellations. The Iridium system moves around the Earth periodically with a duration of 100 min, and considering the inter-satellite links establishment time, only time slots with a duration of 8 s or more are selected for the simulation experiments [19]. Both the Betw strategy and the strategy proposed in this paper set 7 satellite nodes as cache nodes, this experiment assumes that all satellite nodes have the same cache capacity, the users' request process follows the Poisson distribution, and the users' request pattern obeys the Zipf distribution, and the replacement strategy of all nodes is LRU strategy, the specific parameters are set as follows during the experiment.

**Table 1.** Experimental parameter settings

| Parameter | Value |
|---|---|
| Number of users | 22 |
| Number of producers | 1 |
| Default cache capacity | 30 |

(*continued*)

**Table 1.** (*continued*)

| Parameter | Value |
|---|---|
| Zipf parameter α | 0.7–1.3 |
| Default parameter α | 1.0 |
| Request rate/(req·s$^{-1}$) | 50–350 |
| Default request rate/(req·s$^{-1}$) | 200 |
| Number of content | 500–3500 |
| Default number of content | 1000 |

### 5.2 Evaluation Indicators

To evaluate the performance of the CSSI strategy, the cache hit ratio and the average content request delay are used as evaluation metrics.

(1) Cache Hit Ratio.

The cache hit ratio (CHR) is the ratio of the data requests number that are hit by cache nodes to the total number of requests in selected time slots.

$$CHR = c / F \tag{7}$$

where c denotes the number of requests that are hit by cache nodes in the selected time slots, and F denotes the number of all requests in the selected time slots.

(2) Average Content Request Delay.

The average content request delay (ACRD) is the average delay between the time a data request node sends a data request and the time it receives the data.

$$ACRD = \frac{\sum_{i=1}^{F} d_i}{F} \tag{8}$$

where $d_i$ denotes the time to get the data packet for each data request and F denotes the number of all content requests sent by data request nodes.

### 5.3 Experimental Results

This section evaluates the CSSI against LCE, LCD, Prob (0.3), Prob (0.5), Prob (0.7), and Betw in terms of the cache hit ratio and average content request delay.

In Fig. 4, all caching strategies show a significant improvement in the cache hit ratio and a decrease in average content request delay as the Zipf parameter increases, the CSSI consistently has a higher cache hit ratio than the other strategies, with the LCD strategy performing the worst and several other strategies similarly. The average content request delay gradually decreases with increasing the Zipf parameter, and when the Zipf parameter is 1.3, the CSSI has a delay of 816 ms and the worst Betw strategy is 1212.9 ms.

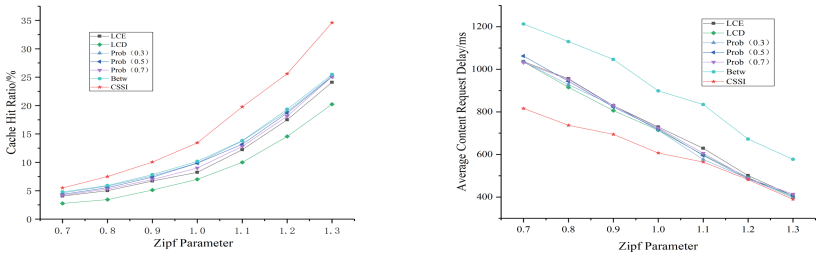



**Fig. 4.** Impaction of Zipf parameter on cache hit ratio and average content request delay.

As the Zipf parameter becomes larger, users' requests for data become concentrated, and the differences between data start to manifest, the more concentrated requested data already cached by satellite nodes can satisfy most data requests. The Betw strategy caches on satellite nodes that occupies the shortest paths, its cache hit effect is better than other strategies, while the satellite network is time-varying, the transmission paths are constantly changing, so it has the worst delay effect. The LCD strategy has the worst cache-hitting effect because the user nodes keep moving throughout the period and the cached contents will keep swinging.

As shown in Fig. 5, with the increase of content request frequency, the cache hit ratio of all caching strategies tends to decrease and the average content request delay tends to increase, the cache hit ratio and average content request delay of CSSI are better than other caching strategies.

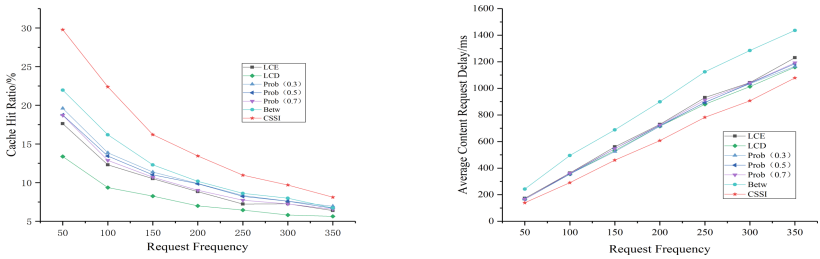



**Fig. 5.** Impaction of request frequency on cache hit ratio and average content request delay.

When the frequency of content requests increases, the number of requests sent by user nodes increases, the popular contents cached in the cache nodes becomes increasingly insufficient to meet the demand, and the cache hit ratio gradually decreases so that the data required by users need to be available at more distant cache nodes or even the data production node, the average content request delay is high. For the cache hit ratio, the changes of Betw and LCD strategies are similar to the reasons for the Zipf parameter. Prob (0.3) caches the content with low probability on all satellite cache nodes but makes the content distribution more uniform compared to Prob (0.5), Prob (0.7), LCE, etc. In terms of average content request delay, the cache nodes selected by the CSSI can return popular contents to more satellite nodes, so the data requests are better satisfied throughout the period and the average content request delay is lower.

As shown in Fig. 6, as the number of content increases, the cache hit ratio decreases, and the average content request delay increases, the decreasing trend is most obvious when the number of content reaches 1000, after which the decreasing trend gradually decreases, and the CSSI has the best performance. When the number of content reaches 1500, the growth rate of average content request delay gradually decreases.

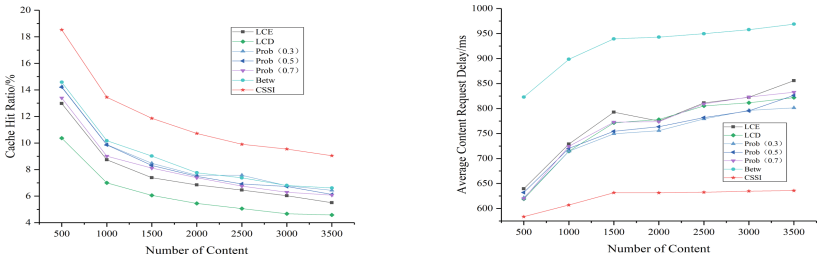



**Fig. 6.** Impaction of number of Content on cache hit ratio and average content request delay.

The CSSI strategy performs well when the number of content is small, but when the number of content increases, the above-mentioned problem will occur, causing the deterioration of the cache hit ratio and content request delay. The Betw strategy uses the transmission path as a reference to find cache nodes, and the time-varying nature of the satellite network causes changes in the transmission path and increases the average content request delay, while the Prob (0.3) strategy outperforms other caching strategies in this respect. The CSSI strategy doesn't involve changes in the packets' delivery paths and returns popular contents to more satellite nodes, users' requests are more easily satisfied, and thus the CSSI achieves a better cache result.

## 6 Summary

ICN makes it easier for data request users to obtain data through in-network caching. To reduce the time-varying effect on data transmission and to improve transmission efficiency among satellites, this paper introduces ICN into the satellite networks and proposes a caching strategy for information-centric satellite networks based on spreading influence. Our strategy decomposes the satellite networks' period into steady topological time slots to solve the satellite mobility problem, calculates the spreading influence of satellite nodes within a single time slot with the k-shell method and information entropy, calculates the global spreading influence with the help of time slot length, and obtains a better cache performance by caching popular contents with dynamic probability in the satellite nodes with high global spreading influence. Simulation experiments show that the proposed caching strategy has a better cache hit ratio and lower average content request delay than LCE, LCD, Prob, and Betw strategies.

## References

1. Li, J., Xue, K., Liu, J., et al.: A user-centric handover scheme for ultra-dense LEO satellite networks. IEEE Wireless Commun. Lett. **9**(11), 1904–1908 (2020)

2. Li, W., Oteafy, S.M.A., Fayed, M., et al.: Quality of experience in ICN: keep your low-bitrate close and high-bitrate closer. IEEE/ACM Trans. Netw. **29**(2), 557–570 (2020)
3. Li, Y., Wang, Y., Yuan, P., et al.: Popularity-aware back-tracing partition cooperative cache distribution for space–terrestrial integrated networks. IET Commun. **13**(17), 2786–2796 (2019)
4. Man, D., Lu, Q., Wang, Y., et al.: An adaptive cache management approach in ICN with pre-filter queues. Comput. Commun. **153**, 250–263 (2020)
5. Laoutaris, N., Syntila, S., Stavrakakis, I.: Meta algorithms for hierarchical web caches. In: IEEE International Conference on Performance, Computing, and Communications, pp. 445–452. IEEE (2004)
6. Gupta, D., Rani, S., Ahmed, S.H., et al.: Edge caching based on collaborative filtering for heterogeneous ICN-IoT applications. Sensors **21**(16), 5491 (2021)
7. Chai, W.K., He, D., Psaras, I., Pavlou, G.: Cache "less for more" in information-centric networks. In: Bestak, R., Kencl, L., Li, L.E., Widmer, J., Yin, H. (eds.) NETWORKING 2012, pp. 27–40. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30045-5_3
8. Cai, Y.P., Liu, J., Fan, X.W.: Node centrality metric based caching mechanism in content-centric network. J. Commun. **38**(06), 10–18 (2017)
9. Li, L., Liu, H.Y., Lu, L.F.: Probabilistic caching content placement method based on content-centrality. J. Comput. Res. Dev. **57**(12), 2648–2661 (2020)
10. Man, D., Lu, Q., Wang, H., et al.: On-path caching based on content relevance in information-centric networking. Comput. Commun. **176**, 272–281 (2021)
11. Zhu, L., Fang, S.L., Hu, Q., et al.: Evaluation method for time-varying satellite topology network node importance. Syst. Eng. Electron. **39**(06), 1274–1279 (2017)
12. Xu, J., Song, T., Yang, Y.T., et al.: Research on caching of multilayered satellite networks. Manned Spaceflight **25**(4), 461–467 (2019)
13. D'Oro, S., Galluccio, L., Morabito, G., et al.: SatCache: a profile-aware caching strategy for information-centric satellite networks. Trans. Emerging Telecommun. Technol. **25**(4), 436–444 (2014)
14. Zhu, L., Ren, Z.Y., Guo, X.B., et al.: Low delay routing strategy based on steady-state satellite network. Radio Commun. Technol. **47**(5), 603–610 (2021)
15. Xu, R., Di, X., He, X., et al.: Evaluation method of node importance in temporal satellite networks based on time slot correlation. EURASIP J. Wirel. Commun. Netw. **2021**(1), 1–23 (2021)
16. Borgia, E., Bruno, R., Passarella, A.: Reliable data delivery in ICN-IoT environments. Futur. Gener. Comput. Syst. **134**, 271–286 (2022). https://doi.org/10.1016/j.future.2022.04.004
17. Mastorakis, S., Afanasyev, A., Zhang, L.: On the evolution of ndnSIM: An open-source simulator for NDN experimentation. ACM SIGCOMM Comput. Commun. Rev. **47**(3), 19–33 (2017)
18. Gomez, C., Darroudi, S.M., Naranjo, H., et al.: On the energy performance of iridium satellite IoT technology. Sensors **21**(21), 7235 (2021)
19. Jiao, Z., Liu, B., Liu, E., et al.: Low-pass parabolic FFT filter for airborne and satellite lidar signal processing. Sensors **15**(10), 26085–26095 (2015)

# Posture and Appearance Fusion Network for Driver Distraction Recognition

Hao Yu[1], Chong Zhao[2,3(✉)], Xing Wei[1,2,4], Yan Zhai[1], Zhen Chen[5], Guangling Sun[6], and Yang Lu[1,4]

[1] School of Computer and Information, Hefei University of Technology, Hefei, China
zhaochong@hfut.edu.cn
[2] Intelligent Manufacturing Institute of Hefei University of Technology, Baohe, China
[3] Engineering Quality Education Center of Undergraduate School, Hefei University of Technology, Hefei, China
[4] Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Beijing, China
[5] School of Computer Science and Technology, Anhui University, Hefei, China
[6] School of Electronic and Information Engineering, Anhui Jianzhu University, Hefei, China

**Abstract.** Distracted driving is the act of driving while engaged in other activities, such as using a cell phone, texting, eating, or reading, which takes the driver' attention away from the road. Nowadays, the distracted driving detection models based on deep learning can extract critical information from video data to characterize the driving behavior process. But the distraction driving method based solely on appearance features cannot essentially eliminate the noise impact of the complex environment on the model, and the distracted driving recognition method based solely on skeletal information is unable to recognize the joint action of the human body and the objects. Therefore, the development of an accurate distracted driving detection model has become challenging. In this paper, we propose a distracted driving recognition model MFD-former based on the fusion of posture and appearance. First, a feature extraction module is proposed to extract skeleton data(i.e., posture) and appearance features(i.e., descriptors), which are merged by a graph neural network. Then, the two kinds of information are input into the MFD-former encoder module, and the self-attention mechanism quickly extracts the sparse data. Finally, the classification results of distracted driving are obtained by extracting the classification labels through the MLP Head. The MFD-former model outperforms existing models. It achieved 95.1% accuracy on the State Farm dataset and 90.24% accuracy on the self-built Train Drivers dataset.

**Keywords:** Driver distraction recognition · Attention mechanism · Graph neural network · Heterogeneous information fusion


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 165–174, 2022.
https://doi.org/10.1007/978-3-031-19208-1_14

## 1 Introduction

Distracted driving is a phenomenon in which the driver's attention is directed to activities unrelated to normal driving (calling, smoking, etc.), which leads to a decline in driving ability. Distractions can multiply driving safety risks and even lead to traffic accidents. In recent years, using computer vision technology to monitor driver behavior and dynamically identify distracted driving has become a research hotspot [9].

The distracted driving recognition method, based on the object detection network VGG and so on [6], roughly divides the human body into different categories of targets, such as head, hand, shoulder, etc., and judges human behavior by its relative position, validated on public datasets such as Kaggle [4] for distracted driving. However, in a specific application environment, it will be affected by conditions such as background and lighting. For example, when a person's clothing color is similar to the background color, it may not be recognized correctly.

Due to the method relying solely on image features, it is still unable to accurately describe the spatiotemporal features of human behavior. Distracted driving recognition methods based on skeletal information, such as ST-GCN [14] and NAS-GCN [10], can separate human gesture recognition from the influence of light, and background, and have strong robustness. However, the appearance features cannot be integrated since the skeleton data is obtained. For example, when the driver uses the same hand to make a phone call or adjust his glasses, the movement of joint points is almost the same, so the two kinds of actions cannot be distinguished correctly.

In order to solve the above problems, this paper proposes **M**ulti-information **F**usion **D**river Activity Recognition network based on Trans**former** [13](MFD-former). First, the MFD-former uses OpenPose [3] to extract the coordinate information of the posture and then uses the Visual Descriptors Extraction module to extract the appearance features of the joint points, that is, the descriptors of the joint points. Then, the graph neural network is introduced to fuse the two types of information, input the fused information into the MFD-former Encoder, and use the self-attention mechanism to quickly extract the sparse data. Finally, the classification labels are extracted by Multilayer Perceptron Head(MLP Head) to obtain the classification results of distracted driving.

We compare MFD-former with other state-of-the-art baselines to highlight the advantages of the proposed approach. Moreover, we study the model at different scales to investigate the impact of the number of parameters and attention heads. To sum up, our contributions are as follows:

1. An attention-based MFD-former model is proposed, demonstrating that self-attentional architectures can outperform existing convolutional and graph convolutional distracted driving recognition models.
2. The graph neural network is used to fuse posture and descriptors, fully mining the driving behavior under the joint description of appearance features and skeletal data.

3. A visual descriptors extraction module is proposed to extract the descriptors in the images.

## 2 Related Work

### 2.1 Graph Neural Network

The graph contains rich information, and many studies have begun to use deep neural network models to learn the feature representation of nodes in the network. Extending deep neural network models to non-Euclidean data, that is, graph convolutional neural networks(GCN), has become an emerging research hotspot [1]. HetGNN [15] obtains different types of neighbor nodes of each node by adopting random walk and restart sampling strategy, and then use different neural network modules to aggregate different characteristics of nodes, the same type of neighbor nodes, and different types of neighbor information, respectively. Finally, the representation of the node is formed. With the ability of these methods to represent graph data, graph heterogeneous graph neural models are also gradually applied to different types of information aggregation. This paper fuses the skeleton data and appearance features by graph neural networks.

### 2.2 Driver Behavior Recognition

Existing deep learning methods for driver distraction recognition mainly focus on two data patterns: Appearance features and Skeleton data.

Appearance features: As Koesdwiady proposed an end-to-end deep learning solution for distracted driving image recognition. The framework utilizes a pre-trained convolutional neural network VGG-19 [6] for feature extraction, and adds two fully connected layers to fine-tune VGG-19. Moslemi proposed to utilize 3D convolutional neural network and optical flow method [8] to improve driver distraction detection tasks to obtain helpful information from temporal information.

Skeleton data: This type of method uses the coordinates of the human skeleton to determine actions and processes the skeleton coordinates to obtain feature values that can be used to perform action recognition. There are two main methods for extracting skeleton data: (1) Using a camera (such as Controller-Pose [2]), (2) using various tools to extract key points from RGB images (such as OpenPose [3]). Maosen Li proposed an action recognition network ST-GCN [14], which uses motion structure graph convolution and temporal convolution as basic building blocks to learn the spatiotemporal characteristics of actions.

## 3 Proposed Method

In this section, we will introduce the architecture of the MFD-former(Fig. 1). We define $V = \{\mathrm{I}_t \in R^{H\times W\times 3}\}_{t=1}^{T}$ as the input, $H$ and $W$ are the height and width of the video, $T$ is the frame number of the input video, and $I_t$ is

the video frame. First, the OpenPose extracts the skeleton data in the picture, and the Visual Descriptors Extraction module extracts descriptors. Two kinds of information are fused by graph neural network, that is, a Patch in Fig. 1 obtained in a frame of video. And a trainable vector Class[token] is added to the input sequence. This vector forces the self-attention in the MFD-former Encoder module to aggregate the information into a compact high-dimensional representation to separate different driver behavior categories. Then we need to input $T+1$ Patches into the MFD-former Encoder module, use the self-attention mechanism to extract the sparse data quickly, and finally use the MLP Head to extract the classification labels.

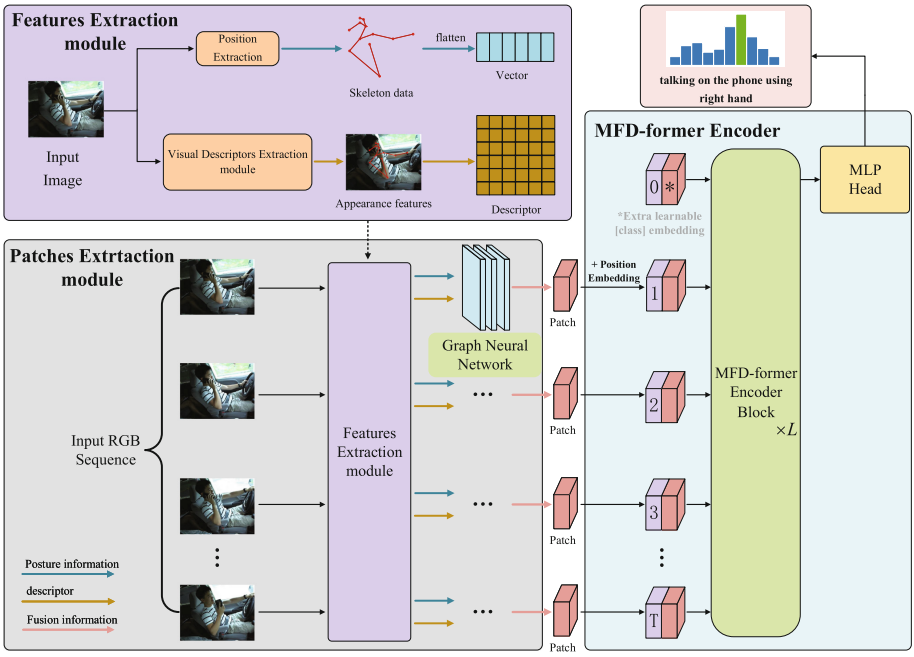



**Fig. 1.** Schematic of our MFD-former. The MFD-former consists of two network models: the Patches Extraction module(gray part) and MFD-former Encoder(blue part). The Patches Extraction module extracts the Patch on each frame of video, that is, the fusion information of the posture and appearance. $T+1$ Patches are input into the MFD-former Encoder module, and sparse data are extracted quickly by the self-attention mechanism, to obtain classification results through MLP Head. $T$ is the frame number of the input video. (Color figure online)

### 3.1 Patch Extraction Module

The purpose of the module is to extract the Patch of each frame of the video, which is mainly composed of the Features Extraction module and the Graph

Neural Network. Features Extraction module consists of the Position Extraction and Visual Descriptors Extraction module.

**Position Extraction Module.** In this article, the Position Extraction module is based on OpenPose [3]. The obtained posture $p_i := (x_i, y_i)$ of the driver represents the abscissa and ordinate of the $i$-th joint point. Distracted driving has nothing to do with the lower body, considering the specific driving scenario. To avoid and reduce the impact of lower limb movements on distracting behavior recognition, we will only consider the driver's upper limb nodes(1. Left ear 2. Left eye 3. Nose 4. Right eye 5. Right ear 6. Left wrist 7. Left elbow 8. Left shoulder 9. Neck 10. Right shoulder 11. Right elbow 12. Right wrist).

**Visual Descriptors Extraction Module.** We propose the Visual Descriptors Extraction module to extract descriptors for each joint. It mainly includes two parts: encoder blocker and decoder blocker. However, we get the descriptor of the whole image here, which is not all we need. We compare the $p_i$ obtained by Position Extraction to find the descriptor $d_i$ we need.

**Graph Neural Network.** The $p_i$ and $d_i$ obtained by the Features Extraction module are two different types of information, so we use the GNN to fusion(Eq. 1). In this paper, $g_i$ is used to represent the fusion result of the posture feature of the $i$-th joint point and the descriptor, and ${}^{(0)}g_i$ represents the initial feature after fusion. Moreover, we use Multilayer Perceptron (MLP) to embed joint point information into a high-dimensional vector; The formula is as Eq. 1.

$$ {}^{(0)}g_i = d_i + MLP_{enc}(p_i) \tag{1} $$

### 3.2 MFD-former Encoder

**Patch Embedded.** What we get through the Graph Neural Network is a two-dimensional matrix. Before entering the MFD-former Encoder, we need to add [class]token and Position Embedding(Eq. 2).

$$ z_0 = [g_{class}; g_i^1 E; g_i^1 E; \cdots; g_i^T E] + E_{pos},\ E \in R^{256\times D}, E_{pos} \in R^{(T+1)\times D} \tag{2} $$

We insert $g_{class}$ for classification into the $T$ tokens we just got, which is a trainable parameter. $E_{pos}$ is the position code added to the original feature, which is related to the frame number in the video.

**MFD-former Encoder Architecture.** MFD-former is to stack the MFD-forme Encoder Block $L$ times repeatedly. The MFD-forme Encoder Block consists of an alternation of Multi-Head Self-Attention (MSA) and MLP blocks.

$$ z'_l = MSA(LN(z_{l-1})) + z_{l-1},\ l = 1 \ldots L \tag{3} $$

Equation 3 is the MSA part, including multi-head self-attention, skip connection (Add) and layer normalization (Norm), which can repeat $L$ times.

$$z_l = MLP(LN(z'_l)) + z'_l,\ l = 1\ ...\ L \tag{4}$$

Equation 4 is the MLP Block part, including feedforward network (FFN), skip connection (Add) and layer normalization (Norm), and can also repeat $L$ times.

$$C = LN(z_L^0) \tag{5}$$

Equation 5 is layer normalization. An MLP Head is used as the classification head lastly, where is the output logit vector of the model, i.e. the classification labels. The other tokens are the only inputs to the module, but the supervision signal only comes from the [CLS] token.

## 4 Experiments

### 4.1 Datasets Settings

**State Farm Dataset.** The State Farm insurance company released a dataset available that classifies images into 10 categories [4]. Although the source dataset is still images, we reconstructed the time-series relationship from the CSV files in the original dataset. We obtained 20094 ten-frame sequences and 20094 ten-frame sequences, and appropriate labels were applied to each sequence.

**Train Drivers Dataset.** We built a specific dataset for train driver driving situations, divided into eight categories, which include two categories of normal driving videos and six categories of distracted driving videos. As shown in Fig. 2. We cut each video into about 3 s, and obtained a total of 9362 instances. To ensure the diversity of our data, we select participants with different heights, weights, and driving styles, wearing different uniforms, as shown in (1) and (2) in Fig. 2, and record videos with different brightness during the day and night, as shown in (1) and (7) in Fig. 2.

### 4.2 Implementation Details

We used the State Farm dataset and the self-built Train Drivers dataset in the following experiments. Moreover, we divide each action video into $T$ frames, and input different versions of MFD-former on the experimental results. The hyperparameter analysis of the three versions of MFD-former is summarized in Table 1.

In experiments, we used the Adam Optimization Algorithm [5] for all training, $\beta_1 = 0.9, \beta_2 = 0.999$, a batch size of 1 and apply a high weight decay of 0.1. $T = 10$ when using the State Farm dataset, $T = 40$ when using the Train Drivers dataset. The ratio of training to test for both datasets is $6 : 4$.


**Fig. 2.** The Train Drivers dataset contains eight classes, including Category 2 Normal Driving and Category 6 Distracted Driving. (2) in the diagram is normal driving, the thumb and little finger of one hand stand up, indicating that the train enters the sideline. The dataset was recorded by multiple drivers in various environments. The video resolution is $1280 \times 720$ 13 Hz.

**Table 1.** Details of MFD-former model variants. Layers are the stacking times of the encoder block; Hidden size is the length of the token vector; MLP size is the number of nodes in the first fully connected layer of the MLP block in the Encoder block; Heads are Multi-head Attention the number of heads in.

| Model | Layers | Hidden size D | MLP size | Heads | Params |
|---|---|---|---|---|---|
| Base | 12 | 768 | 3092 | 12 | 86M |
| Large | 24 | 1024 | 4096 | 16 | 307M |
| Huge | 32 | 1280 | 5120 | 16 | 632M |

### 4.3 Comparison Results

**Results on State Farm Dataset.** We chose four posture-based(Pos) models and three appearance-based(App) models, and all seven models have timing information. The experimental results are shown in Table 2. The baseline in the experiment is the traditional skeleton recognition ST-GCN model [14]. Posture-Based model data preprocessing uses OpenPose to extract skeleton data. It can be seen that the accuracy of our model has been greatly improved compared to the baseline. The results of ST-GCN are not ideal because it only relies on posture and ignores the importance of appearance features. These results validate the validity and rationality of our consideration of appearance features. Through the experimental results, we can see that the Ours-Huge model is better than the Ours-Large and Ours-Base model.

**Results on Train Drivers Dataset.** Table 3 shows the results of different methods on the Train Drivers dataset, selecting ST-GCN as the baseline. Our proposed method significantly outperforms baseline because ST-GCN which only relies on posture has certain limitations, and our method uses appearance features. We also have some improvements to the C3D method which only relies

**Table 2.** Comparisons with state-of-the-art driver distraction recognition methods on the State Farm dataset. "Pos" and "App" denote posture and appearance, respectively.

| Model | Modality | Accuracy[%] |
|---|---|---|
| BaseLine | Pos | 77.4 |
| 2S-AGCN [12] | Pos | 88.5 |
| NAS-GCN [10] | Pos | 89.5 |
| ST-TR [11] | Pos | 92.5 |
| C3D, Sports 1M pre-training [7] | App | 73.3 |
| I3D-RGB, ImageNet + Kinetics pre-training [8] | App | 91.8 |
| I3D-two stream, ImageNet + Kinetics pre-training [8] | App | 94.4 |
| Ours-Base | App + Pos | 92.5 |
| Ours-Large | App + Pos | 94.3 |
| Ours-Huge | App + Pos | **95.1** |

on appearance, so it is clear that our method is generally effective. The results of the three models proposed in this paper on the Train Drivers dataset are still the Ours-Huge method with the highest accuracy.

**Table 3.** Comparisons with state-of-the-art driver distraction recognition methods on the Train Drivers dataset. "Pos" and "App" denote posture and appearance, respectively.

| Model | Modality | Accuracy[%] |
|---|---|---|
| Baseline | Pos | 75.21 |
| NAS-GCN [10] | Pos | 88.73 |
| C3D, Sports 1M pre-training [7] | App | 85.51 |
| Ours-Base | App + Pos | 88.52 |
| Ours-Large | App + Pos | 89.42 |
| Ours-Huge | App + Pos | **90.24** |

## 5 Conclusion

In this paper, we propose an MFD-former model to complete the classification of drivers' driving behavior to solve the problem of driver distraction detection. We explore the application of Transformer to driving behavior recognition research. We have verified the feasibility of our model on public datasets and our proposed Train Drivers dataset through multiple sets of experiments, and the accuracy is

not lower than the current advanced time-series methods. In future work, we will continue to study the problem of driving behavior recognition, and try to improve the driving recognition speed under the high recognition accuracy.

**Acknowledgement.** This work was supported by Joint Fund of Natural Science Foundation of Anhui Province in 2020 (2008085UD08), Anhui Provincial Key R&D Program (202004a05020004), Open fund of Intelligent Interconnected Systems Laboratory of Anhui Province (PA2021AKSK0107), Intelligent Networking and New Energy Vehicle Special Project of Intelligent Manufacturing Institute of HFUT (IMIWL2019003, IMIDC2019002).

## References

1. Abadal, S., Jain, A., Guirado, R., López-Alonso, J., Alarcón, E.: Computing graph neural networks: a survey from algorithms to accelerators. ACM Comput. Surv. (CSUR) **54**(9), 1–38 (2021)
2. Ahuja, K., Shen, V., Fang, C.M., Riopelle, N., Kong, A., Harrison, C.: Controllerpose: inside-out body capture with VR controller cameras. In: CHI Conference on Human Factors in Computing Systems, pp. 1–13 (2022)
3. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7291–7299 (2017)
4. Farm, S.: State farm distracted driver detection. Technical report (2016). https://www. kaggle. com/c/state . . .(2016)
5. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
6. Koesdwiady, A., Bedawi, S.M., Ou, C., Karray, F.: End-to-End deep learning for driver distraction recognition. In: Karray, F., Campilho, A., Cheriet, F. (eds.) ICIAR 2017. LNCS, vol. 10317, pp. 11–18. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59876-5_2
7. Lemley, J., Bazrafkan, S., Corcoran, P.: Transfer learning of temporal information for driver action classification. In: MAICS, pp. 123–128 (2017)
8. Moslemi, N., Azmi, R., Soryani, M.: Driver distraction recognition using 3d convolutional neural networks. In: 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA), pp. 145–151. IEEE (2019)
9. Moslemi, N., Soryani, M., Azmi, R.: Computer vision-based recognition of driver distraction: a review. Concurrency Comput.: Pract. Experience **33**(24), e6475 (2021)
10. Peng, W., Hong, X., Chen, H., Zhao, G.: Learning graph convolutional network for skeleton-based human action recognition by neural searching. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 2669–2676 (2020)
11. Plizzari, C., Cannici, M., Matteucci, M.: Skeleton-based action recognition via spatial and temporal transformer networks. Comput. Vis. Image Underst. **208**, 103219 (2021)
12. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12026–12035 (2019)
13. Vaswani, A., et al.: Attention is all you need. In: Advances in neural information processing systems, vol. 30 (2017)

14. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
15. Zhang, C., Song, D., Huang, C., Swami, A., Chawla, N.V.: Heterogeneous graph neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 793–803 (2019)

# A Behavior Decision Method for Autonomous Vehicles in an Urban Scene

Jiujun Cheng[1(✉)], Yonghong Xiong[1], Shuai Feng[2], Guiyuan Yuan[1], Qichao Mao[1], and Bo Lu[3]

[1] The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai, China
{chengjj,2032960,gyuan_yuan,2010476}@tongji.edu.cn
[2] China Unicom Smart Connection Technology Ltd., Beijing, China
fengshuai2@chinaunicom.cn
[3] Tianhua College Shanghai Normal University, Shanghai, China
lb2364@sthu.edu.cn

**Abstract.** Autonomous vehicles sense the surrounding environment through various sensors and make behavior decisions based on real-time perception information to change their vehicle's motion state. Most existing studies on behavior use single data, high computational complexity, and single optimization criteria only, which lacks practicality. This work proposes an autonomous vehicle motion behavior decision method. It first extracts the corresponding features according to correlation among adjacent vehicles and predicts driving behavior and trajectory of adjacent vehicles. Then, it abstracts driving states of autonomous vehicles, introduces their state transition process based on a definite state machine, and gives a behavior decision method. Finally, a multi-objective optimization algorithm is used to optimize. Extensive simulation results show that this method can effectively improve the safety, efficiency, and practicability of autonomous vehicle motion behavior decision.

**Keywords:** Autonomous vehicle · Trajectory prediction · Motion behavior decision · Evolutionary algorithm

## 1 Introduction

With the development of artificial intelligence, autonomous vehicle technology is developing rapidly. Autonomous vehicles sense their surroundings through communication [1] and various sensors and guide their motion behavior through decision methods.

Deep-learning-based methods are widely used in existing studies on behavior decision of autonomous vehicles. It is highly dependent on data sets, and the original data used can hardly represent the environment around the vehicle. And the existing studies have a single optimization standard, mainly considering safety, efficiency, and other goals, without too much consideration of practicality.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 175–184, 2022.
https://doi.org/10.1007/978-3-031-19208-1_15

In this work, we firstly predict behavior and trajectory of adjacent vehicles based on the vehicle's attributes and the correlation among adjacent vehicles. Then we propose an autonomous vehicle motion behavior decision method based on a definite state machine and optimize it through a multi-objective optimization algorithm. The key contributions of our work are as follows:

1) We abstract influencing factors around the adjacent vehicle of an autonomous vehicle, extract higher-dimensional feature information, and predict behavior and trajectory of adjacent vehicles.
2) We introduce state transition process based on definite state machine and propose a method for behavior decision of autonomous vehicles.
3) We use a multi-objective optimization algorithm to optimize the motion behavior decision method.

Our work can improve the safety, efficiency, and practicability of autonomous vehicles' motion behavior decisions, making it possible for autonomous vehicles to be applied in complex scene.

## 2 Trajectory and Behavior Prediction Method

### 2.1 Surrounding Vehicle Collection Extraction

Firstly, we fill and filter the adjacent vehicle dataset. Then, we extract and process the original features, including relative features and drive behaviors. The lanes in the dataset are generally two-way two-lane, but only vehicles in nearby lanes have a direct impact on the target vehicle. So in the process of feature extraction, only vehicles in adjacent lanes are considered.

After extracting vehicle features, in order to reduce the interference of some features with large numerical distribution on the model and accelerate the convergence speed, Z-score standardization is carried out for it.

Then, according to the LK-DBSCAN (Limit-K-DBSCAN) algorithm, we extract the set of vehicles that affect the vehicles selected for the study from the data. LK-DBSCAN algorithm firstly divides the whole data set according to the sampling time point to obtain $Data_t$, and extracts the sampling set *Set(t)* corresponding to each time point. On this basis, traverse all vehicles $i$ in *Set(t)*, calculate the distance $U_{ki}$ from other vehicles $k$ and put it into set $U_k$. After the complete distance set $U_i$ is calculated, the vehicle set $ArgMin_i^t$ that satisfies the constraint $Min(\varepsilon, \beta, U_i)$ is solved. $Min(\varepsilon, \beta, U_i)$ is described as follows:

$$Min(\varepsilon, \beta, U_i) = \{k, ..., l \,|U_{ki} \leq \varepsilon \ \wedge \ k \in Top_{\min -\beta}(U_i)\} \tag{1}$$

where $\varepsilon$ is the "visua" perception distance of autonomous, $\beta$ is the upper limit of the number of adjacent vehicle that affect autonomous.

### 2.2 Behavior and Trajectory Prediction of Adjacent Vehicles

First, we use the predicted vehicle's current surrounding environment information and its historical behavior as model input, and extract the information through LSTM. Then, the SoftMax function is used to calculate the probability values of the three behaviors of keeping straight, changing lanes left, and changing lanes right.

As the problem is a multi-classification problem, we select the cross-entropy function as loss function. And the loss in a single batch is described as follows:

$$L = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{3} y^{(i,k)} \log h_{\theta,k}(I^{(i)}) \tag{2}$$

where $m$ represents the quantity of samples in each batch, $y^{(i,k)}$ represents the truth label of the *k-th* behavior of sample *i*, $h_{\theta,k}(I^{(i)})$ represents the probability of the *k-th* behavior predicted by the model.

We batch the input by time and train the model to get the weight. Then, we design an encoder and decoder composed of LSTM cells. We take the prediction of the adjacent vehicle's behavior as the influencing factor for trajectory prediction. The encoder reuses the characteristics of the relationship between vehicles and the historical trajectory of the vehicle. The output of the encoder and the result of behavior prediction are combined as the input of the decoder, and then the adjacent vehicle trajectory information is output.

## 3 Autonomous Vehicle Behavior Decision Model

### 3.1 Formal Specification

**Definition 1:** The stop state $w$ which includes the autonomous vehicle initialization state and the state where the speed is 0 for a long time. It is defined as follows:

$$\begin{gathered} \hat{s} = w \\ s.t.\ \hat{s} = \text{initialized} \ \vee \ \mathrm{S_t} = 0(\mathrm{t} = 1, 2, ..., \mathrm{k}), \end{gathered} \tag{3}$$

where $\hat{s}$ represents the current state of autonomous vehicle.

**Definition 2:** The following state $f$ is:

$$\begin{gathered} \hat{s} = f \\ s.t. \begin{cases} \exists\ S_a, |S - S_a| \leq \alpha \ \wedge \ \|P - P_a\|_2 \geq \beta \\ \gamma \leq S \leq \delta \end{cases} \end{gathered} \tag{4}$$

where $S_a$ and $P_a$ respectively represent the speed and position of the car ahead, $\alpha$ and $\beta$ respectively represent the threshold of relative velocity and relative distance.

**Definition 3:** If there is no vehicle in front of the driving lane, this state can be defined as a free state $r$ due to the reduction of influencing factors.

**Definition 4:** The overtaking state of $o$ is defined as follow:

$$s.t. \begin{cases} \hat{s} = o \\ S_a(t) < S(t) \leq \delta \\ C = \text{true} \\ 0 \leq A(t) \leq \hat{A} \\ P_a(k).y - P(k).y = 0 \\ t = 1, 2, ..., k \end{cases} \tag{5}$$

**Definition 5:** The lane change state $b$ is defined as follow:

$$s.t. \begin{cases} \hat{s} = b \\ \gamma \leq S(t) \leq \delta \\ 0 \leq A(t).x \leq \hat{A} \\ t = 1, 2, ..., k \end{cases} \tag{6}$$

### 3.2 Autonomous Driving Behavior Decision

According to five driving states, we give the constraints of state transition:

1) If the current state of autonomous vehicle is $w$, it can make the following transition:

$$\begin{cases} w \to r & V_a = \emptyset \\ w \to f & \text{otherwise} \end{cases} \tag{7}$$

where $\hat{v}_i$ and $\hat{V}_a$ respectively represent the autonomous vehicle and the set of vehicles ahead.

2) If the current state of autonomous vehicle is $f$, it can make the following transition:

$$\begin{cases} f \to o & d > \mu\beta \ \wedge \ S_a < S_t \ \wedge \ \hat{s}_r \neq o \\ f \to r & \hat{V}_a = \emptyset \\ f \to b & l_i \in \hat{L}_o \ \wedge \ \left\| \hat{t}(t) - \hat{t}_r(t) \right\| \geq \beta, t = 1, ..., k \\ f \to f & \text{otherwise} \end{cases} \tag{8}$$

where $\hat{L}_c$, $\hat{L}_o$ and $\hat{l}_i$ respectively represent the set of carriageways, the set of overtaking lanes, and the lane where test autonomous vehicle drive, $\mu$ is the safety distance coefficient and $\beta$ is the safe distance, and $\hat{t}$ represents the trajectory of the vehicle.

3) If the current state of autonomous vehicle is $r$, it can make the following transition:

$$\begin{cases} r \to b & l_i \in \hat{L}_o \ \wedge \ \left\|\hat{t}(t) - \hat{t}_r(t)\right\| \geq \beta, t = 1, ..., k \\ r \to f & V_a \neq \emptyset \\ r \to w & d \leq \beta \ \wedge \ \hat{r} \\ r \to r & \text{otherwise} \end{cases} \tag{9}$$

where $\hat{r}$ represents whether the traffic light is red.

4) If the current state of autonomous vehicle is $o$, it can make the following transition:

$$\begin{cases} o \to f & l_i \in \hat{L}_c \ \wedge \ \hat{V}_a \neq \emptyset \\ o \to r & l_i \in \hat{L}_c \ \wedge \ \hat{V}_a = \emptyset \\ o \to o & \text{otherwise} \end{cases} \tag{10}$$

5) If the current state of autonomous vehicle is $b$, it can make the following transition:

$$\begin{cases} b \to f & l_i \in \hat{L}_c \wedge \hat{V}_a \neq \emptyset \\ b \to r & l_i \in \hat{L}_c \wedge \hat{V}_a = \emptyset \\ b \to b & \text{otherwise} \end{cases} \tag{11}$$

Based on the state transition, we then propose a Definite Machine of Autonomous Vehicles (DMAV) for driving behavior decision. First, according to the current state $\hat{s}$, we get the state set $\hat{s}_n$ that it can be converted to. Then put the states in $\hat{s}_n$ that satisfy the constraint $\hat{C}_{\hat{s} \to \hat{s}_n}$ into $\hat{s}_p$. Jump to different states $\hat{s}_t$ in $\hat{s}_p$ according to random seeds, and set different lateral acceleration and longitudinal acceleration.

The calculation equation of acceleration consists of the following three parts:

1) Influence coefficient of expected velocity $W_t$:

$$W_t = (\log_{S_t}(S_t - S + 1))^2 \tag{12}$$

2) Influence coefficient of safety distance $W_s$:

$$W_s = -e^{D_s - (P_a - P)} \tag{13}$$

where $D_s$ is the safety distance.

3) Influence coefficient of relative speed with front vehicle $W_{Vel}$:

$$W_v = \frac{S_a - S}{Max(S_a, S) + 1} \tag{14}$$

To sum up, the calculation equation of acceleration can be represented as:

$$F(S_t, S, S_a, P_a) = a * (W_t + W_s + W_v) \tag{15}$$

where $a$ represents the maximum acceleration of the autonomous vehicle.

### 3.3 Multi-objective Evolutionary Algorithm for DMAV

1) *Objective Function:* In an autonomous vehicle behavior decision model, we need to consider both the efficiency, security, and practicability of a decision. The optimal solution is a good trade-off between them. To this end, the following three maximization objective functions are respectively given:

$$\hat{E} = \tilde{S}(T) \text{ or } \hat{\mathrm{E}} = \tilde{\mathrm{T}}(\mathrm{s}), \tag{16}$$

where $\hat{E}$ represents the efficiency of autonomous vehicle behavior decision, $\tilde{S}(T)$ represents the distance traveled in time $t$, and $\tilde{T}(s)$ represents the time required for a distance of $s$.

$$\hat{S} = \{\left|P_i^t - P_a^t\right| | t \in (0, 1, ..., T) \wedge if\, no\, P_a^t, P_a^t = \infty\}, \tag{17}$$

where $\hat{S}$ represents the security of autonomous vehicle behavior decision, $|P_i^t - P_{ahead}^t|$ represents the distance between autonomous vehicle $i$ and the vehicle in front at time $t$, when there is no preceding vehicle, this parameter has no meaning, so set it to infinity.

$$\hat{U} = \{v_t - v_{t-1} | t = (0, 1, 2, ..., k)\}, \tag{18}$$

where $\hat{U}$ represents practicability of autonomous vehicle behavior decision, and $\{v_t - v_{t-1} | t = (0, 1, 2, ..., k)\}$ represents the fluctuation of acceleration in a continuous period of time.

2) *Constraint Conditions:* According to the above three objective functions, we can divide constraints into static constraints and dynamic constraints.
   a) *Static constraints:* It mainly includes the physical constraints $K$ of the autonomous vehicle and Road environmental constraints $G$.

$$k_s : \langle K, G \rangle, \tag{19}$$

   b) *Dynamic constraint:* Since autonomous vehicles make behavior decisions in a dynamic environment, there are some constraints that change with the state of it. The most representative one is the safety distance. There is a dynamic constraint on the safety distance $\hat{D}_i^t$:

$$k_d : \hat{D}_i^t = \tilde{D} + v_t * \Delta t + \frac{1}{2} * \left|\tilde{A}\right| * \left\lceil \frac{v_t}{\left|\tilde{A}\right|} \right\rceil^2 \tag{20}$$

where $\tilde{D}$ is the minimum distance between the front and rear of the vehicle when the speed is 0.

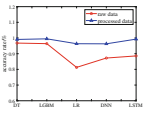
3) *Optimal solution:* Under the above constraint conditions, the autonomous vehicle behavior decision model is formulated as:

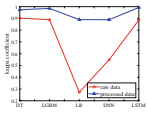
$$\begin{cases} ArgMax(\hat{E} \,|\, \hat{E}_i = \tilde{S}(t)) \text{ or } \mathrm{ArgMin}(\hat{\mathrm{E}}_\mathrm{i} = \tilde{\mathrm{T}}(\mathrm{s})) \\ ArgMax(\hat{S}_i) \\ ArgMin(\hat{U}_i) \\ s.t.\ k_s \ \wedge \ k_d \end{cases} \tag{21}$$

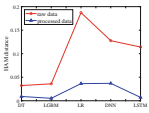
The autonomous vehicle behavior decision method only based on security and efficiency does not consider the weight of each influencing factor and the impact on practicability. Therefore, by adding the weight vector, we rewrite the calculation equation of acceleration $\dot{F}$ as:

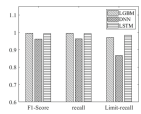
$$\dot{F} = a * (w_1 * W_t + w_2 * W_s + w_3 * W_v). \tag{22}$$

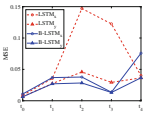
Then we optimize the solution of $[w_1, w_2, w_3]$. Firstly, the weight coefficient $\boldsymbol{V}_1$ is added with *random* respectively, and the simulation experiment is carried out to calculate $\hat{D}$ and the distance from the car ahead $D^*$. If $D^*$ is greater than or equal to $\hat{D}$ at any time, it is considered that the behavior decision method strictly meets the safety requirements under this weight coefficient, put $[w_1, w_2, w_3]$ into $\boldsymbol{V}_t$. Then, $\boldsymbol{V}_r$ is traversed. $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ are weighted and combined, and $\boldsymbol{V}_t$ is updated after experiment and calculation. Finally, in order to approximate the optimal solution space, sort $\boldsymbol{V}_t$ according to practicality, and update the top 60% of practicality to $\boldsymbol{V}_r$, and then continue the iteration.

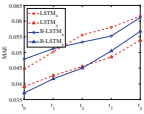

(a) Accuracy.

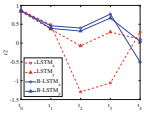

(b) Kappa consistency coefficient

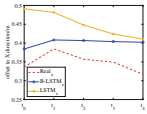

(c) HAM distance


(d) Comparison of different algorithms


(e) MSE


(f) MAE


(g) R2


(h) Offset in X direction

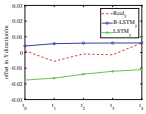

(i) Offset in Y direction

**Fig. 1.** Trajectory prediction results.

## 4 Experiments and Evaluation

### 4.1 Experiment Results

We first compare LSTM to DecisionTree (DT), Light GBM (LGBM), Logistic Regression(LR), and Deep Neural Network (DNN) on two different input data:

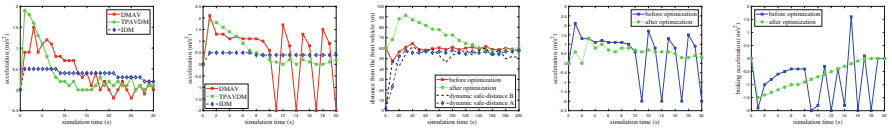
1) Raw data 2) Relevant feature data extracted by modeling the association relationship of adjacent vehicles.

Figure 1(a)-(c) respectively show the accuracy, kappa consistency coefficient, and HAM distance of each method on two different input data. In the model with relevant feature data, the performance reflected by the three indicators of the five-vehicle behavior prediction methods has been improved. And the vehicle behavior prediction method based on LSTM improved significantly, mainly because the under-fitting problem is solved by adding more extracted features.

Figure 1(d) shows the comparison between LSTM and LGBM and DNN after adding the relevant feature data as input. The F1-Score and Recall of DNN are lower than those of the other two algorithms, and the Limit-Recall of LSTM is higher than LGBM while the other two indicators are flat, which indicates that the LSTM has a better performance in processing time-series data.

Then, we compare B-LSTM to the traditional LSTM-based vehicle trajectory prediction method(LSTM). Figure 1(e) shows the MSE of the LSTM and B-LSTM methods in vehicle driving direction and its vertical direction. And X indicates the direction of vehicle travel and Y indicates the vertical direction of vehicle travel. Compared with LSTM, B-LSTM is more stable in the MSE, especially in the X direction, which indicates B-LSTM can more accurately predict the vehicle trajectory, and the prediction of vehicle behavior can significantly reduce the error of the model in the X direction. Figure 1(f) shows the MAE(Mean Absolute Error) of LSTM and B-LSTM. B-LSTM is lower than LSTM in more time, it can be seen that B-LSTM the prediction effect of B-LSTM is better. Figure 1(g) shows the R2 coefficient of determination of LSTM and B-LSTM. At time T0-T4, compared with LSTM, the mean value of B-LSTM is closer to 1 and the fluctuation is smaller in both X direction and Y direction, which indicates that the fitting effect of B-LSTM is better and more stable.

Figure 1(h) and Fig. 1(i) show the offset curves of the two methods in the X direction and Y direction of a single sample, respectively. The curve fitted by B-LSTM is closer to the real vehicle trajectory, especially $B - LSTM_x$, which verifies adding the prediction of vehicle behavior can effectively reduce the fitting error of the model.




(a) Acceleration (b) Acceleration (c) Distance from the front vehicle (d) Acceleration (e) Braking acceleration

**Fig. 2.** Behavior decision results

### 4.2 Simulation Results of Behavior Decision Model

We use SUMO for the behavior decision experiments. In order to better simulate the interference of adjacent vehicles to autonomous vehicles, CACC [2] algorithm is adopted to control adjacent vehicles. We compare the safety, efficiency, and practicability of DMAV, TPAVDM, and IDM under the following state through a simulation experiment.

Figure 2(a) shows the acceleration of autonomous vehicles with time under the condition of fixed safety distance. Compared with TPAVDM, the acceleration of DMAV is relatively stable in the start-up stage. And compared with the other two methods, the acceleration of DMAV can be stabilized to zero faster, so DMAV has better practicability.

Figure 2(b) shows the acceleration of autonomous vehicles with time under the condition of fixed safety distance. Compared with TPAVDM and IDM, DMAV has larger acceleration fluctuation and poor convergence, so it performs poorly in practicability.

Figure 2(c) shows the distance between the autonomous vehicle and the preceding vehicle and the dynamic safety distance before and after the optimization of DMAV. After optimization, the distance curve first increases and then decreases until it fluctuates with the dynamic safety distance. The main reason is that to improve practicability, DMAV makes concessions in efficiency after optimization, but with the increase of simulation time, the distance curve after optimization is gradually close to that before optimization. After a multi-objective optimization algorithm, DMAV has higher security than before optimization when ensuring the same efficiency.

Figure 2(d) and Fig. 2(e) show the acceleration of autonomous vehicles before and after optimization of DMAV under the following state and braking state. Compared with before optimization, the acceleration fluctuation of the optimized DMAV algorithm is smaller and shows a downward trend. Therefore, the optimized DMAV has higher practicability.

## 5 Conclusion

Different from previous studies, this paper introduces a multi-objective optimization-based behavior decision method for autonomous vehicles. It simplifies driving states of autonomous vehicles, gives the behavior decision method, and optimizes it from the aspects of safety, efficiency, and practicality. Our future work should proceed in the following aspects:

1) The introduction of an autonomous vehicle network can help autonomous vehicles better perceive the surrounding environment. [3–5] are instructive.
2) Building vehicle groups [6] may further reduce the computational complexity.
3) Privacy protection [7] is very important and can effectively improve safety.

**Acknowledgements.** This work was supported in part by the NSFC under Grant 61872271 and 62272344, in part by the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) under Grant SKLNST-2020-1-20.

## References

1. Wang, C., Chen, C., Pei, Q., Jiang, Z., Xu, S.: An information centric in-network caching scheme for 5G-enabled internet of connected vehicles. IEEE Trans. Mob. Comput. (2021). https://doi.org/10.1109/TMC.2021.3137219
2. Xiao, L., Wang, M., Schakel, W., van Arem, B.: Unravelling effects of cooperative adaptive cruise control deactivation on traffic flow characteristics at merging bottlenecks. Transp. Res. Part C: Emerg. Technol. **96**, 380–397 (2018)
3. Cheng, J., et al.: Accessibility analysis and modeling for IoV in an urban scene. IEEE Trans. Veh. Technol. **69**(4), 4246–4256 (2020)
4. Cheng, J.J., Yuan, G.Y., Zhou, M.C., Gao, S.C., Huang, Z.H., Liu, C.: A connectivity prediction-based dynamic clustering model for VANET in an urban scene. IEEE Internet Things J. **7**(9), 8410–8418 (2020)
5. Cheng, J., Yuan, G., Zhou, M., Gao, S., Liu, C., Duan, H.: A fluid mechanics-based data flow model to estimate VANET capacity. IEEE Trans. Intell. Transp. Syst. **21**(6), 2603–2614 (2019)
6. Cheng, J.J., et al.: A dynamic evolution method for autonomous vehicle groups in a highway scene. IEEE Internet Things J. **9**(2), 1445–1457 (2021). https://doi.org/10.1109/JIOT.2021.3086832
7. Zhang, J., Cui, J., Zhong, H., Chen, Z., Liu, L.: PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks. IEEE Trans. Dependable Secure Comput. **2**, 722–735 (2021)

# TimeBird: Context-Aware Graph Convolution Network for Traffic Incident Duration Prediction

Fuyong Sun[1], Ruipeng Gao[1], Weiwei Xing[1(✉)], Yaoxue Zhang[2(✉)], Wei Lu[1], Jun Fang[3], Shui Liu[3], Nan Ma[3], and Hua Chai[3]

[1] Beijing Jiaotong University, Beijing 100044, China
{fysun12,rpgao,wwxing,luwei}@bjtu.edu.cn
[2] Tsinghua University, Beijing 100084, China
zhangyx@tsinghua.edu.cn
[3] DiDi Corporation, Beijing 100089, China
{fangjun,liushui,mandymanan,chaihua}@didiglobal.com

**Abstract.** Estimating the traffic incident duration is of great importance to traffic control, traffic navigation, and transportation safety. However, the complex road network topology and dynamic traffic conditions make it challenging. In this paper, we propose a context-aware spatio-temporal graph convolution framework, named **TimeBird**, to estimate the duration time of traffic incidents. Specifically, we build the dynamic weighted adjacency matrix and traffic incident risk similarity matrix to learn the hidden spatial context correlations based on graph convolution network. Then we employ the historical traffic speed of road segments to learn the temporal dependency. Lastly, we design a context-aware attention mechanism to adaptively learn the heterogeneous traffic features for incident duration prediction. Extensive experiments on two large-scale real-world datasets from DiDi ride-hailing platform demonstrate the effectiveness of TimeBird.

**Keywords:** Traffic incident duration · Spatio-temporal features · Graph convolution network

## 1 Introduction

Traffic incidents are common in our daily transportation owing to various factors, such as bad weather and aggressive driving behaviors. If traffic incident occurs, it will have a severe impact on road conditions of multiple adjacent road links. Therefore, if the duration of traffic accident can be predicted timely and accurately, Traffic Management Systems (TMS) can take effective measures to

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant No. 2021YJS185, in part by National Natural Science Foundation of China under Grant No. 62072029 and No. 61876017, in part by Beijing NSF under Grant No. L192004, and in part by DiDi Research Collaboration Plan.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 185–195, 2022.
https://doi.org/10.1007/978-3-031-19208-1_16

reduce losses, and ride-hailing platforms (e.g., DiDi, Uber) can present the bubble trip reminder or give alternate routes for users to eliminate the side-effect of traffic incident.

Traffic incident duration is defined as the time gap from its occurrence to its ending in traffic scenarios. As shown in Fig. 1, it consists of three components, discovery time (between occurrence and reporting), response time (between reporting and starting of clearance) and clearance time (between the starting of clearance and the ending). In recent years, there are several studies [1–3] on estimating the duration of traffic incidents. However, they still have some challenges in capturing the heterogeneous spatio-temporal correlations, such as local similarity and global tendency in road network topology, external factors, like rush hours, holidays and weekdays etc.

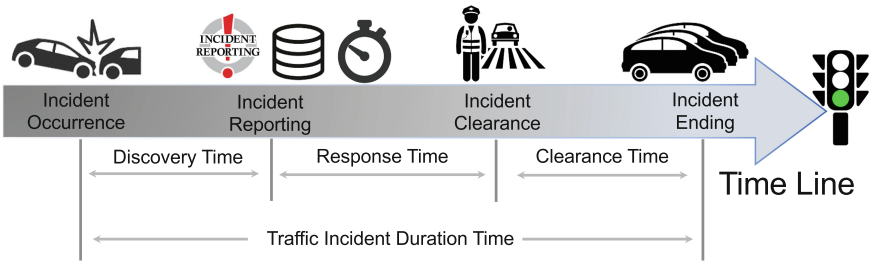



**Fig. 1.** The time of traffic incident duration.

To address above challenges, we propose a context-aware spatio-temporal graph convolution network for traffic incident duration prediction, called TimeBird. Our contributions are summarized as follows:

- A spatio-temporal framework is proposed to capture the traffic patterns and road network topological dependency. Specifically, we build dynamic weighted adjacency matrix in road network and risk similarity matrix to learn the spatial representations of each road segment.
- We design a context-aware attention mechanism to adaptively learn the heterogeneous traffic features for incident duration prediction.
- Extensive experiments are conducted on two large-scale real-world traffic datasets in Beijing and Shanghai, collected by DiDi ride-hailing platform, which demonstrate the effectiveness of proposed model.

## 2 Related Work

### 2.1 Spatio-temporal Forecasting

In recent years, deep neural networks and graph neural networks have achieved a significant performance in spatio-temporal forecasting, such as traffic flow forecasting, estimated time of arrival and risk prediction, etc. Specially, DCRNN [4]

proposed diffusion convolutional recurrent neural network to capture the spatio-temporal dependencies for traffic forecasting. DeepSTTE [5] integrated the CNN and TCN to estimate the short-term travel time. Multi-graph [6] and SCEG [7] utilized the spatio-temporal features to predict the bike flow and bike-sharing demand. [8–11] designed spatio-temporal learning framework to estimate the customized travel time and traffic accident risk respectively, based on heterogeneous traffic features. However in scenario of predicting traffic incident duration, we also need to consider the risk level of road segments except as the spatiotemporal relationship.

### 2.2 Traffic Incident Duration

The existing methods on incident duration prediction are mainly divided into two categories. One is data-driven approaches [1,2,12], they mostly regarded the incident duration prediction as a regression task based on traffic sensor data. [2] classified the congested level and sequentially predicted the duration of each level. [12] proposed a gradient boosting decision trees (GBDTs) model to predict the incident clearance time. Another is deep learning methods [3, 13,14]. Specifically, [13] designed an attention-based framework to learn traffic patterns for traffic incident impact forecasting separately. However, in urban road network, road segments have local similarity and global periodicity in traffic behaviors. Therefore, we propose a context-aware fusion model to capture the dynamic dependencies of road network for traffic incident duration prediction.

## 3 Problem Definition

**Definition 1.** ***Road Network.*** *We treat the road network as a directed graph* $G = (V, E, A, F, X)$. *Specifically,* $V$ *is set of nodes (road segments) and* $E$ *is the set of edges (connectivity between road links) in the graph.* $A \in \mathbb{R}^{N \times N}$ *is adjacency matrix, where* $N$ *is the number of nodes.* $F \in \mathbb{R}^{\mathbb{N} \times \mathbb{M}}$ *indicates the feature matrix of each road segment in road network, where* $M$ *is the feature dimensions of each road segment. It mainly includes length, width, direction, road class (e.g., high-speed), number of lanes and speed limit. We denote the traffic matrix as* $X \in \mathbb{R}^{\mathbb{N} \times \mathbb{D}}$, *where* $D$ *is the time dimension. For example, the element* $x_i^t$ *of* $X$ *indicates the observed traffic speed of* $i$*-th link at time step* $t$.

**Traffic Incident Duration:** Given an incident occurrence time $t_o$, the location where the accident occurred (i.e., the road link), and the current traffic speed $x_i^{t_o}$ of the road link, our aim is to predict the duration of the traffic accident with combining the heterogeneous spatio-temporal features and contextual information.

The variable parameters are summarized in Table 1.

## 4 Model Architecture

To capture the traffic fluctuation and estimate the duration time of traffic accident, we propose the spatio-temporal framework, named TimeBird, to learn the heterogeneous hidden features about traffic accidents. The overall architecture of TimeBird is shown in Fig. 2. It mainly includes three subcomponents, i.e., spatial correlations, temporal dependency and contextual information. Specifically, we explore graph convolution networks to capture the hidden topology in spatial correlations, then utilize GRU to learn the temporal dependency from the historical traffic speed, and lastly employ embedding method to transform the external attributes (e.g., week, holiday) into low dimensional vector. To adaptively model the heterogeneous features, we utilize the attention mechanism to calculate the attention weights for predicting the traffic accident duration.

**Table 1.** The description of notations

| Variables | Description |
|---|---|
| $R_i^t$ | Indicates the sums of traffic accidents occurred in road link $i$ at time interval $t$ |
| $\hat{A}$ | Normalized adjacency matrix |
| $S_r^i$ | The road properties distributions of road link $i$ in road network $r$ |
| $H_t$ | The hidden state at time interval $t$ |
| $H_a$ | The hidden features with attention weights |

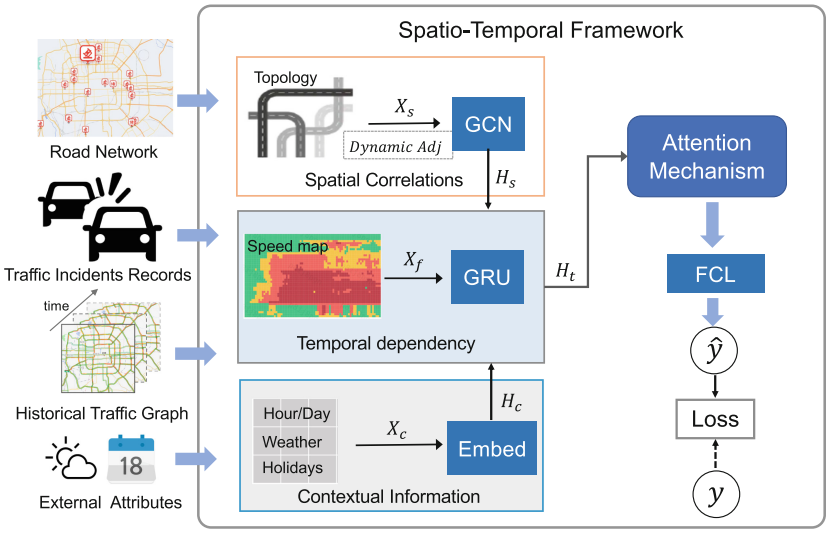



**Fig. 2.** The architecture of TimeBird. We explore the spatio-temporal learning components and attention mechanism for traffic incident duration prediction.

### 4.1 Spatial Correlations

The post-effect of a traffic accident is strongly affected by the hidden road network topology and road characteristics. To adaptively model the spatial correlations, we build the dynamic adjacency matrix and risk similarity matrix, then employ GCN to learn the representation of each road segment.

Specially, the great success of graph convolution have demonstrated in capturing graph data. According to the [15], the hidden layer of GCN encodes both graph structure and features of nodes, which can be formulated as follows:

$$\begin{aligned} H^{l+1} &= \sigma(\hat{A}H^{(l)}\theta^{(l)}) \\ \hat{A} &= \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}} \\ \widetilde{A} &= A + I \end{aligned} \tag{1}$$

where $\sigma(\cdot)$ is an activation function. $\theta^{(l)}$ means the trainable weight matrix. $H^{(l)}$ means the hidden matrix of $l$-th layer, and $H^{(0)} = X$, which means the feature matrix. $\widetilde{A}$ denotes the adjacency matrix with self-loops. $I$ is the identity matrix, $\widetilde{D}$ is the degree matrix, which is calculated as $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$.

To capture the spatial correlations of road links from different aspects, we build two similarity matrices, including dynamic adjacency matrix and risk similarity matrix. Specifically, we extract the road segments and traffic accident risk between any two road links and utilize the Jensen-Shannon divergence [10,16] to calculate the similarity weights $A_* = \{A_{road}, A_{risk}\}$. Taking the road similarity as an example, the method is as follows:

$$A_{road}(i,j) = \begin{cases} 1 & \text{if road link } i \text{ and road link } j \text{ are adjacent} \\ e^{-JS(S_r^i || S_r^j)} & \text{otherwise} \end{cases} \tag{2}$$

$$JS(S_r^i, S_r^j) = \frac{1}{2} \sum_{1 \le d \le D} \left( S_r^i(d) \log \frac{2S_r^i(d)}{S_r^i(d) + S_r^j(d)} + S_r^j(d) \log \frac{2S_r^j(d)}{S_r^i(d) + S_r^j(d)} \right)$$

where $S_r^i, S_r^j \in \mathbb{R}^D$ mean the road properties distribution of road link $i$ and $j$ in road network topology $r$. $S_r^i(d)$ denotes the $d$-th dimension of $S_r^i$. Similarly, we utilize traffic accident risk $R_i^t$ to calculate the risk similarity $A_{risk}(i,j)$ in different time intervals.

Then we utilize a two-layer graph convolution to produce the hidden features of road segment, with considering dynamic spatial correlations and traffic risk. The forward of graph convolution can be presented as follows:

$$H_s = f(X, A_*) = \sum_{*\in\{road,risk\}} \sigma(A_* \sigma(A_* X W_*^{(0)}) W_*^{(1)}) \tag{3}$$

where $\sigma(\cdot)$ means the activation function, such as $ReLU$, $W_*^{(\cdot)}$ is the learnable weight matrix. $H_s$ denotes the output of the multi-graph convolutions for capturing the spatial correlations.

### 4.2 External Attributes

The occurrence of traffic accidents is strongly correlated with external factors, such as weather, working days, morning and evening rush hours, etc.

For example, as shown in Fig. 3, we plot the hourly ration fluctuation of traffic accidents occurrence during one day in weeks. The $x$ axis is the hour of one day, $y$ axis denotes the ratio of traffic accident occurrence in each day. Specifically, there are high incidence in morning and evening rush hour. In addition, the working days and weekends have different pattern of traffic fluctuation.

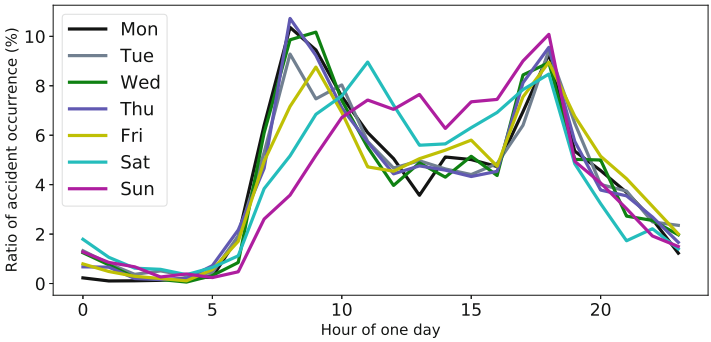



**Fig. 3.** The fluctuation of traffic accident occurred in one week.

In this work, we incorporate the heterogeneous features of external attributes for predicting traffic incident duration. Owing to these features are categorical value, they cannot be directly fed into neural networks. Therefore, we utilize the embedding method [8] to transform them into low-dimensional vectors as $H_c$. In our experiment, we embed the day in one week into $\mathbb{R}^7$, morning or evening rush hour into $\mathbb{R}^3$, weather condition into $\mathbb{R}^6$, and the holiday into $\mathbb{R}^2$.

### 4.3 Temporal Correlations

The occurrence of traffic accidents has short-term similarity during one day and long-term periodicity in weeks. In Fig. 3, we can observe that there is an adjacent trend and week periodicity during one day and weeks. In addition, the traffic condition of road segment is a crucial factor for estimating traffic accident duration. Therefore, we utilize time series model GRU to capture the temporal correlations for forecasting with integrating the traffic speed of road segment.

Specifically, we extract the traffic speed data from the recent $T$ time intervals and the same time interval in the historical $p$ weeks to produce the sequential features $X_f = (X_p, X_{t-T+1}, ..., X_t), f = p + T$ as the input to capture the temporal correlations:

$$H_t = GRU\Big([H_s||X_f||H_c], H_{t-1}\Big) \tag{4}$$

where || is vector concatenation operation at specific dimension. $H_t$ means the hidden state at time interval $t$ with considering the spatio-temporal correlations and contextual information.

### 4.4 Context-Aware Attention Mechanism

In traffic scenarios, different road conditions at various time periods have different effects on accident duration prediction. Inspired by [10], we utilize an attention mechanism to adaptively model the dynamic spatio-temporal features. The hidden feature with attention weights $H_a$ is calculated as follows:

$$H_a = \sum_{t=1}^{s} \alpha_t \cdot H_t \tag{5}$$

where $t$ is the time interval, $s$ is the length of time window. $\alpha_t$ is an attention weight at $t$-th time step, $H_t$ is the hidden feature of the output in GRU, and the sum of attention weight is $\sum_{t=1}^{s} \alpha_t = 1$.

Specifically, the attention weight $\alpha_t$ is calculated as:

$$\alpha_t = softmax\Big(ReLU(W_t \cdot H_t + b_t)\Big) \tag{6}$$

where $W_t$ and $b_t$ are learnable parameters.

### 4.5 Model Training

Owing to the traffic accidents are affected by various factors, so we fuse the spatio-temporal features and external features to learn the correlations and accurately predict the duration time of accident. We feed it into the FC (Fully-Connected Layer) to dynamically capture the relation of hidden features.

$$\hat{y} = FC\Big(W_a \cdot H_a + b_a\Big) \tag{7}$$

where $H_a$ denotes the heterogeneous features with attention weights, $W_a$ is the learnable weight matrix, $b_a$ is the bias item.

In model training phase, we utilize mean-squared error (MSE) as the loss function to learn the model parameters, which is calculated as follows:

$$Loss(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{8}$$

where $y$ is the ground truth of traffic accident duration, $\hat{y}$ denotes the prediction of proposed model, $i$ is the $i$-th training sample, $N$ is the total number of training data.

## 5 Experiments

In this section, we evaluate the performance of TimeBird on two large-scale real traffic datasets from DiDi ride-hailing platform. We present the details as follows:

### 5.1 Implementation

The time period of traffic datasets is from Nov 1th to Nov 28th, 2021. It mainly includes the traffic event data, road network topology, road characteristics, traffic conditions and external informations. The sample of traffic event records the time, location, accident type, and duration of the accident. As for the road characteristics (e.g., length, width), owing to the various data range, we employ Z-score to normalize it in data preprocessing. The details of Parameters and Evaluation Metrics are as follows:

**Parameters:** In experiment, we divide the dataset into training, validation and test set with a ratio of 6:2:2. To learn the temporal correlation, we set the time interval as 2 min. The hidden size of GRU is 128. During the model training phase, we utilize the Adam optimizer and set the learning rate as $lr = 0.001$. We implement the TimeBird based on PyTorch, and train the model on 64-bit server with NVIDIA GTX 2080Ti.

**Evaluation Metrics:** To validate the performance of TimeBird on traffic incident duration, we adopt three criterions, MAPE(Mean Absolute Percentage Error), MAE (Mean Absolute Error), and RMSE (Rooted Mean Square Error).

### 5.2 Comparison with Baseline Methods

To evaluate the effectiveness of TimeBird, we compare it with the following baselines:

**SVR** [17]: SVR (Support vector regression) is widely utilized in time-series analysis and statistical learning.

**RF** [2]: Random Forest is a method to predict the post-impact after the traffic accidents using the crowdsourcing data.

**ASTGCN** [18]: An attention-based spatio-temporal graph convolution network is designed for traffic flow forecasting. Because of the different forecasting tasks, we convert multi-step into single-step prediction in experiments.

**HastGCN** [13]: It proposes a hierarchical attention-based spatio-temporal model for predicting the impact of traffic incidents from traffic sensor data.

**Table 2.** The comparisons for traffic accident duration prediction.

| Methods | Beijing | | | Shanghai | | |
|---|---|---|---|---|---|---|
| | MAPE(%) | MAE(min) | RMSE(min) | MAPE(%) | MAE(min) | RMSE(min) |
| SVR | 63.26 | 49.25 | 57.16 | 71.32 | 56.83 | 77.38 |
| RF | 58.42 | 43.58 | 49.63 | 67.02 | 52.31 | 58.06 |
| ASTGCN | 45.17 | 33.69 | 43.26 | 55.81 | 48.79 | 59.24 |
| HastGCN | 40.06 | 34.72 | 41.86 | 47.35 | 38.28 | 43.65 |
| **TimeBird** | **37.52** | **29.31** | **34.28** | **42.93** | **36.33** | **41.82** |

We conduct the experiments on large-scale traffic dataset in Beijing and Shanghai. The experimental results are shown in Table 2. We observe that the MAPE of conventional methods (SVR, RF) is nearly 15% higher than deep learning models (variants of GCN). This is because the duration of traffic accident is not only affected by the road conditions at the accident's occurrence time, but also by the hidden topology of its road segment. Due to the proposed model of TimeBird simultaneously considers the spatio-temporal dependencies and risk similarity, so it is nearly average 3% lower than HastGCN.

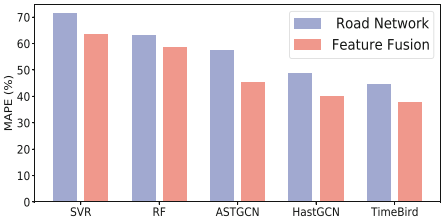



**Fig. 4.** Ablation study on different features.

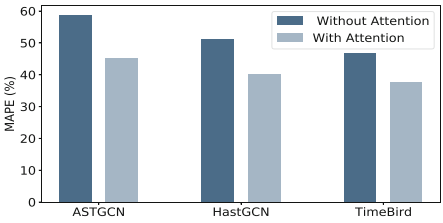



**Fig. 5.** Ablation study on attention mechanism

**Effect of Feature Fusion.** To demonstrate the efficacy of spatio-temporal features, we separately conduct the experiment with different features, like road network topology and feature fusion (combining road network topology and traffic speed). And the result is shown in Fig. 4. We observe that the MAPE of feature fusion is lower than road network in compared methods. Therefore, it can achieve a better performance for incident duration prediction.

**Effect of Attention Mechanism.** To validate the performance of attention mechanism, we conduct the experiments with/without attention mechanism, respectively. The result is shown in Fig. 5. We observe that the deep learning methods with attention achieves a lower value of MAPE than without it.

## 6 Conclusion

In this paper, an end-to-end context-aware spatio-temporal framework is proposed to predict the duration of traffic accident. It effectively models the spatio-temporal correlations with considering the urban road network topology and risk of traffic accident. We have conducted extensive experiments on real-world heterogeneous traffic dataset in two cities, and the results demonstrate the effectiveness of proposed model. However, traffic incident duration is affected by various factors, like the level of traffic congestion and its propagation behaviors. In the future work, we will take it into account to improve the accuracy.

## References


1. Valenti, G., Lelli, M., Cucina, D.: A comparative study of models for the incident duration prediction. Eur. Transp. Res. Rev. **2**(2), 103–111 (2010). https://doi.org/10.1007/s12544-010-0031-4
2. Lin, Y., Li, R.: Real-time traffic accidents post-impact prediction: based on crowdsourcing data. Accid. Anal. Prev. **145**, 105696 (2020)
3. Fu, K., Ji, T., Zhao, L., Lu, C.T.: Titan: a spatiotemporal feature learning framework for traffic incident duration prediction. In: Proceedings of the 27th ACM SIGSPATIAL, pp. 329–338 (2019)
4. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
5. Fu, L., Li, J., Lv, Z., Li, Y., Lin, Q.: Estimation of short-term online taxi travel time based on neural network. In: Yu, D., Dressler, F., Yu, J. (eds.) WASA 2020. LNCS, vol. 12385, pp. 20–29. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59019-2_3
6. Chai, D., Wang, L., Yang, Q.: Bike flow prediction with multi-graph convolutional networks. In: Proceedings of the 26th ACM SIGSPATIAL, pp. 397–400 (2018)
7. Wang, Q., Guo, B., Ouyang, Y., Shu, K., Yu, Z., Liu, H.: Spatial community-informed evolving graphs for demand prediction. In: Dong, Y., Ifrim, G., Mladenić, D., Saunders, C., Van Hoecke, S. (eds.) ECML PKDD 2020. LNCS (LNAI), vol. 12461, pp. 440–456. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-67670-4_27
8. Gao, R., et al.: Aggressive driving saves more time? Multi-task learning for customized travel time estimation. In: IJCAI, pp. 1689–1696 (2019)
9. Moosavi, S., Samavatian, M.H., Parthasarathy, S., Teodorescu, R., Ramnath, R.: Accident risk prediction based on heterogeneous sparse data: new dataset and insights. In: Proceedings of the 27th ACM SIGSPATIAL, pp. 33–42 (2019)
10. Wang, B., Lin, Y., Guo, S., Wan, H.: GSNet: learning spatial-temporal correlations from geographical and semantic aspects for traffic accident risk forecasting. In: Proceedings of the AAAI, vol. 35, pp. 4402–4409 (2021)
11. Gao, R., Sun, F., Xing, W., Tao, D., Fang, J., Chai, H.: CTTE: customized travel time estimation via mobile crowdsensing. IEEE Trans. Intell. Transp. Syst. (2022)
12. Ma, X., Ding, C., Luan, S., Wang, Y., Wang, Y.: Prioritizing influential factors for freeway incident clearance time prediction using the gradient boosting decision trees method. IEEE Trans. Intell. Transp. Syst. **18**(9), 2303–2310 (2017)
13. Fu, K., Ji, T., Self, N., Chen, Z., Lu, C.T.: A hierarchical attention graph convolutional network for traffic incident impact forecasting. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 1619–1624. IEEE (2021)
14. Cong, H., Chen, C., Lin, P.S., Zhang, G., Milton, J., Zhi, Y.: Traffic incident duration estimation based on a dual-learning Bayesian network model. Transp. Res. Rec. **2672**(45), 196–209 (2018)
15. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
16. Zhou, Z., Wang, Y., Xie, X., Chen, L., Liu, H.: RiskOracle: a minute-level citywide traffic accident forecasting framework. In: Proceedings of the AAAI, vol. 34, pp. 1258–1265 (2020)

17. Wu, C.H., Ho, J.M., Lee, D.T.: Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. **5**(4), 276–281 (2004)
18. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI, vol. 33, pp. 922–929 (2019)

# The Link Awareness Driven Resource Allocation Algorithm Based on Scenario Marking and Vehicle Clustering in VANETs

Bixun Zhang[1,2], Xu Ding[1,2(✉)], Hang Zheng[1,2], Xiang Zheng[1,2], and Pengfei Xu[1,2]

[1] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China
dingxu@hfut.edu.cn

[2] Institute of Industry and Equipment Technology, Hefei University of Technology, Hefei, China

**Abstract.** In this paper, we investigate the channel resource allocation problem in device-to-device (D2D) based VANETs. According to the vehicle density, we first mark the urban transportation scenario into intensive and sparse areas, in which we categorize the communication links as "altruistic" and "ego" links respectively in the consequence of marking results and vehicle attributes. Secondly, the altruistic links are further grouped in terms of an improved spectral clustering algorithm proposed hereby. Moreover, channel resources are dedicated to ego links and different clusters of altruistic links in order to alleviate communication interference and achieve better performance. We formulate an optimization problem of power control for channel resource allocation to maximize the total channel throughput. Fortunately, after reshaping the original problem into a D.C (difference of two convex functions) problem, which can be solved by interior point method, the optimal power allocation method is yielded. Intensive simulations are carried out across various configurations, and the results prove that our scheme has superior performance.

**Keywords:** VANETs · Scenario marking · Vehicle clustering · Resource allocation

## 1 Introduction

Vehicular ad hoc network (VANET), as one of the key technologies for realizing intelligent transport systems, is gradually becoming a hot topic of research. Supported by the 3rd Generation Partnership Project (3GPP) and widely deployed

Supported by the Fundamental Research Funds for the Central Universities of China (NO. PA2021GDSK0095).


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 196–205, 2022.
https://doi.org/10.1007/978-3-031-19208-1_17

cellular networks, VANETs are able to provide a promising solution to user messaging aided by device-to-device (D2D) communications [1]. Nonetheless, ensuring the diverse quality of service (QoS) in a complex and changing vehicle network is still a challenging problem [2]. To solve such problems, efficiently allocating shared spectrum resources becomes of paramount importance in VANETs.

Most of the current research is based on allowing D2D users to share the spectrum resources of cellular users, and reinforcement learning is involved as a powerful tool for this purpose. In [3], Ye et al. considered each V2V link as an agent and used reinforcement learning methods to make optimal decisions about the spectrum and transmission power. He et al. combined graph convolutional neural networks (GNN) with reinforcement learning to develop a distributed GNN-based reinforcement learning scheme [4].

In addition to the above-mentioned reinforcement learning schemes, in [5], Zhang et al. applied graph theory to the resource allocation problem in VANETs and proposed resource sharing schemes for both interference-aware and interference-classification graphs. In [6], Hu et al. used the social attributes of vehicle users to develop a socially aware clustering resource allocation algorithm to optimize the transmission power of all links.

In this paper, we firstly go deep in the modeling issue in V2V communications within the scope of urban scenario. More interference exists in places with higher vehicle density, depending on the density of vehicles, we mark the urban scenario as dense and sparse areas and cluster the V2V links in the dense ones. We endow the "altruistic-awareness" and "ego-awareness" to vehicles and links in dense and sparse areas. In this way we can ensure that the interference suffered by links located in sparse areas during resource allocation is always at a low level, thus ensuring fairness in resource allocation. Later, we allocate spectrum resources and control power usage according to the marking results and link attributes. The contributions of this paper are summarized as follows:

- In accordance with different levels of interference in various scenarios, we propose a density peak-based scenario marking scheme to differentiate users. This will ensure fairness in resource allocation and avoid users in low interference scenarios from being assigned to resources with higher interference.
- We model the resource allocation problem as an optimization problem to maximize channel throughput and develop a link clustering method that can effectively reduce interference. To improve the resource efficiency of the VANET, we divide the channel resource competitors into two parts, i.e., clusters of V2V links and individual V2V links, and allocate channels to them individually.
- For the pragmatic implementation of our algorithm and alleviation of network overheads, we develop a distributed power control scheme that allocates power to the links within each channel individually. Through recasting the prime problem into a new form with identical optimality, the optimal allocation solution can be acquired through interior point method.

## 2 System Model

As shown in Fig. 1, we consider a vehicle network with $M$ cellular users (CUE) and $K$ pairs of device-to-device (D2D) users, where all devices are equipped with one single antenna. Each cellular user communicates with the Base Station (BS) to form a V2I link which is used to support a variety of bandwidth-intensive applications. Each pair of D2D users establishes a V2V link to enable safety information to be shared between neighboring vehicles.

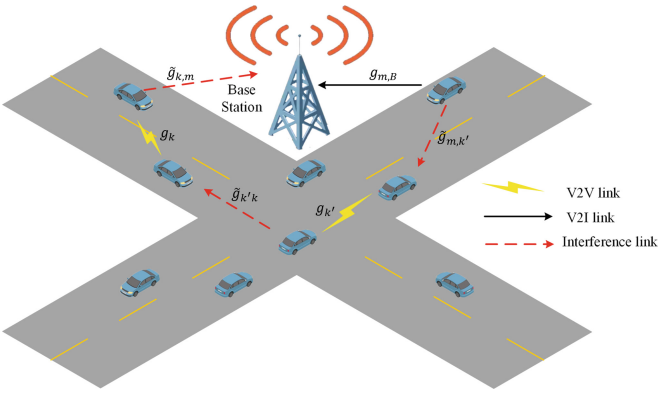



**Fig. 1.** An example of vehicle communications

Define the set of V2I links as $M = \{1, 2, \ldots, m\}$, and the set of V2V links as $K = \{1, 2, \ldots, k\}$. We assume that each V2I link has been pre-assigned an orthogonal spectrum band, i.e., the $m^{th}$ V2I link occupies the $m^{th}$ channel. The set of channels can therefore also be denoted by $M$. To improve the spectrum efficiency of the vehicular network, we allow all V2V links to share spectrum with V2I links, i.e., one or multiple V2V links can share the channels of V2I users.

The signal to interference noise ratio (SINR) for the $m^{th}$ V2I link can be written as

$$r_m^c = \frac{P_m^c g_{m,B}}{\sum_{k=1}^{K} \rho_k^m P_k^d \tilde{g}_{k,m} + \sigma^2} \tag{1}$$

where $P_m^c$ and $P_k^d$ denote the transmission power of the $m^{th}$ V2I link and the $k^{th}$ V2I link respectively, $g_{m,B}$ is the channel gain of the $m^{th}$ V2I link on the $m^{th}$ channel, $\tilde{g}_{k,m}$ is the interference gain of the $k^{th}$ V2I link to the $m^{th}$ V2I link. $\sigma^2$ is the additive white gaussian noise. $\rho_k^m$ is a binary variable, If the $k^{th}$ V2I link uses the channel of the $m^{th}$ V2I link, then $\rho_k^m = 1$, otherwise $\rho_k^m = 0$.

The signal to interference noise ratio (SINR) for the $k^{th}$ V2V link can be written as

$$r_k^d = \frac{P_k^d g_k}{\sum_{k \neq k'}^{K} \rho_{k'}^m \rho_k^m P_{k'}^d \tilde{g}_{k',k} + \sum_{m=1}^{M} \rho_k^m P_m^c \tilde{g}_{m,k} + \sigma^2} \tag{2}$$

where $g_k$ denotes the channel gain of the $k^{th}$ V2V link, and $\tilde{g}_{k',k}$ denotes the interference gain of the $k'^{th}$ V2V link to the $k'^{th}$ V2V link. $\tilde{g}_{m,k}$ denotes the interference gain of the $m^{th}$ V2I link to the $k^{th}$ V2V link. Similarly, $\rho_{k'}^m$ and $\rho_k^m$ are binary variables. $\rho_{k'}^m \rho_k^m = 1$ means that the $k'^{th}$ V2V link shares the same channel with the $k^{th}$ V2V link, and the $k^{th}$ V2V link shares the same channel with the $m^{th}$ V2I link.

Hence the throughput of the $m^{th}$ V2I link is

$$R_m^c = log(1 + r_m^c) \tag{3}$$

The throughput of the $k^{th}$ V2V link is

$$R_k^d = log(1 + r_k^d) \tag{4}$$

In general, V2V links mainly bear vitally important information, such as safety related data, while V2I links support entertainment services of less importance [7]. To ensure quality-of-service (QoS) in V2X networks, the transmission of V2V links should be given a higher priority. Therefore, our goal is to maximize the total throughput of V2V links while ensuring that the V2I links meet the desirable rates. The channel resource allocation problem can be formulated as the following optimization problem:

$$\max_{\rho} R = \sum_{k \in K} log_2(1 + r_k^d) \tag{5}$$

subject to

$$r_m^c \geq r_0 \tag{6}$$

$$0 \leq P_k^d \leq P_{max}, \quad \forall k \in K \tag{7}$$

$$\rho_k^m \in \{0, 1\}, \quad \forall k \in K, m \in M \tag{8}$$

$$\sum_{m \in M} \rho_k^m = 1 \tag{9}$$

where (6) indicates the minimum SINR to be satisfied for each V2I link and (7) indicates the range of transmission power allowed for each V2V user. (8) and (9) show that the channel occupied by each V2I link can be shared by multiple V2V pairs, but only one channel can be occupied by each V2V pair.

Problem (5) is an MINLP problem, and it is difficult to obtain a global optimal solution in a complex and variable V2X network. Therefore, we split it into two subproblems, channel allocation and power control, to solve them individually.

## 3 Channel Resource Allocation Mechanism

This section discusses the clustering and channel allocation scheme for V2V links. The scheme consists of three main components:

### 3.1 Scenario Marking Based on the Density of Vehicles

To filter out V2V links for clustering, the map was marked into "dense areas" and "sparse areas" based on the number and the density of vehicles. Assume that the set of all vehicles in the network is $V = \{v_1, v_2, \ldots, v_i\}$. Inspired by the density peaks clustering algorithm (DPCA) [8], for each vehicle $v_i$ we introduce the local density $\rho_i$ and the relative distance $\delta_i$:

$$\rho_i = \sum_{j \neq i} \chi(d_{ij} - d_c) \tag{10}$$

where $d_{ij}$ is the distance between $v_i$ and $v_j$, and $d_c$ is the truncated distance. $\chi$ is a logical judgment function, which has the expression $\chi(x) = \begin{cases} 1, & x \leq 0 \\ 0, & x > 0 \end{cases}$.

$$\delta_i = \begin{cases} \max\limits_{j}(d_{ij}), & \text{if } \rho_i \text{ is the maximum} \\ \min\limits_{j:\rho_j > \rho_i}(d_{ij}), & \text{otherwise} \end{cases} \tag{11}$$

According to these two functions, we set the criteria for vehicles in dense or sparse areas. The exact steps are shown in Algorithm 2.

In order to ensure the fairness of resource allocation, we give "altruistic awareness" to vehicles in dense areas and "ego awareness" to vehicles in sparse areas. At the same time, we consider the V2V links of vehicles with "altruistic awareness" as "altruistic links" and the other V2V links as "ego links". By differentiating the links, we can ensure that the "ego links" are always subject to a lower level of interference in the resource allocation process.

### 3.2 V2V Links Clustering

Interference amongst V2V links could be more intensive due to the higher density of vehicles, and vice versa. For fairness in resource allocation, we first cluster the V2V links that have a "altruistic awareness" and ensure that interference within each cluster is minimal.

Inspired by graph theory, we consider each V2V link with "altruistic awareness" as a vertex in the graph, with an edge connected them when the vertices interfere with each other. Assume that there is interference between vertex $v$ and vertex $u$, the weight of the edge between them is

$$\omega_{u,v} = \alpha \frac{1}{I_{u,v}} + \beta \frac{1}{I_{v,u}} \tag{12}$$

where $I_{u,v}$ is the interference from the transmitter $u$ to the receiver $v$ and $I_{v,u}$ is the interference from the transmitter $v$ to the receiver $u$. $\alpha, \beta$ are normalization constants.

Based on the calculated weights, we can construct the adjacency matrix $\mathbf{W}$ and the degree matrix $\mathbf{D}$ for the whole graph. Then we use the spectral clustering algorithm to group the nodes into clusters.

**Algorithm 1.** Scenario marking

**Input:** Vehicles collection $D$; Truncation ratio $t$; $k$
**Output:** Collection of vehicles in sparse area $\Psi$;
Collection of vehicles in dense areas $\Phi$

```
Calculate and Rank distances between vehicles;
Use t to find the truncation distance d_c
for i : D do
    Calculate ρ_i
    Normalized to ρ_i
end for
for i : D do
    Calculate δ_i
    Normalized to δ_i
    if δ_i/ρ_i > k then
        Add i to the set Ψ;
    else
        Add i to the set Φ;
    end if
end for
```

### 3.3 Channel Allocation

In this sequel, we divide the competitors of channel resources into two categories: V2V "ego links" and clusters consisting of V2V "altruistic links". The whole process of allocating channel resources is divided into two steps:

(1) First, we calculate the interference gain between V2V link clusters and V2I links in the channel. We subsequently solve this weighted bipartite graph matching problem using the Hungarian algorithm.
(2) For each "ego link", we calculate the interference value of each channel to it and select the channel with the least interference. In this process, we allow two or more ego links to select the same channel.

## 4 Distributed Power Control Scheme

To reduce network overhead, this paper adopts a distributed power control scheme to allocate power to multiple links sharing the same channel to ensure that the channel throughput is at an optimal level.

### 4.1 Power Control Based on the Interior Point Method

Assume that the set of V2V links in the channel is $K_m = \{1, 2, \ldots, K_m\}$. For each channel optimization problem can be rewritten as:

$$\max_{P_{k_m}^d} R_m = \sum_{k \in K_m} log_2(1 + r_{k_m}^d) \tag{13}$$

subject to

$$r_m^c \geq r_0 \tag{14}$$

$$0 \leq P_{k_m}^d \leq P_{max}, \quad \forall k_m \in K_m \tag{15}$$

It is obvious that the objective function is neither convex nor linear, which makes this problem extremely difficult to solve. We first rewrite the problem in the form of the difference between two convex functions:

$$R_m(P) = f_m(P) - l_m(P) \tag{16}$$

where $f_m(P) = \sum_{k_m=1}^{K_m} log_2(\sum_{n=1}^{K_m} P_n^d g_{n,k_m} + P_m^c g_{m,k_m} + \sigma^2)$ and $l_m(P) = \sum_{k_m=1}^{K_m} log_2(\sum_{n \in K_m, n \neq k_m}^{K_m} P_n^d g_{n,k_m} + P_m^c g_{m,k_m} + \sigma^2)$. Problem (13) is equivalent to

$$R_m(P, P') = f_m(P) - (l_m(P') + \langle \bigtriangledown l_m(P'), P - P' \rangle) \tag{17}$$

subject to (14), (15)

To solve (17), we can use the interior point method to approximate the optimal solution by iteration. We introduce the barrier function:

$$\begin{aligned} \phi(P) = &- \sum_{k_m=1}^{K_m} log(P_{k_m}^d) - \sum_{k_m=1}^{K_m} log(P_{max} - P_{k_m}^d) \\ &-log(P_m^c g_{m,B}/r_0 - \sigma^2 - \sum_{k_m=1}^{K_m} P_{k_m}^d g_{k_m,m}) \end{aligned} \tag{18}$$

At this point, the entire optimization problem can be rewritten as:

$$min \quad -tR_m(P, P') + \phi(p) \tag{19}$$

subject to

$$r_m^c \geq r_0 \tag{20}$$

$$0 \leq P_{k_m}^d \leq P_{max}, \quad \forall k_m \in K_m \tag{21}$$

Next, we solve the problem (19) using Newton's iterative method to finally obtain the optimal power allocation for each link in the channel. The detailed process is given in Algorithm 2.

### 4.2 Complexity Analysis

The interior point method is divided into internal and external iterations. The external iteration modifies the dual gap, while the internal iteration finds the appropriate step size using Newton's method. Calculating the step length first requires solving the gradient as well as the Hessian matrix, so the complexity of solving the step length is $O(N^4)$. Assume that the number of internal iterations, is $C_1$, and the number of external iterations is $C_2$. The outermost iteration of Algorithm 2 is $C_3$. Then the overall complexity of Algorithm 2 is $O(C_1 C_2 C_3 N^4)$, where the specific value of $C_1, C_2, C_3$ is controlled by the solving tolerance $\varepsilon$.

**Algorithm 2.** Power Control

**Input:** initial power $P^0$, number of iterations $i$, tolerance $\varepsilon > 0$
**Output:** optimum power $P^i$, channel throughput $R_m^i$
1: **for** $i = 1, 2, \ldots$ **do**
2: Using the interior point method to solve Problem (19) and obtain $P^i$
3: Bring $P^i$ into (13) to calculate the channel throughput $R_m^i$
4: **end for** Until $\mid R_m^i - R_m^{i-1} \mid < \varepsilon$

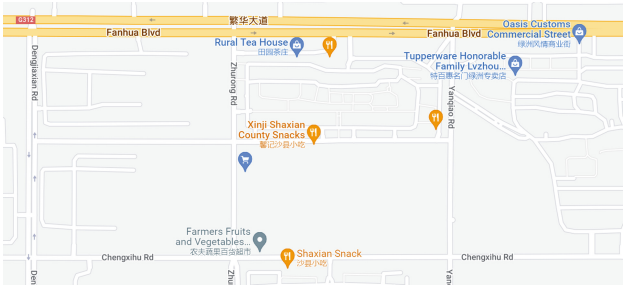



**Fig. 2.** The real road network in a certain area of Hefei, China

## 5 Experiment

To evaluate the performance of the developed resource allocation scheme, the following simulations are carried out. As shown in Fig. 2, a rectangular area of 1800 m × 600 m is captured from the real road network as an urban road scenario, where the base station is located in the center of the area. The specific simulation parameters are shown in Table 1.

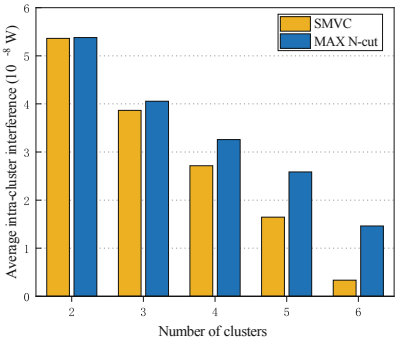



**Fig. 3.** Effect of the number of clusters on the average intra-cluster interference

Figure 3 compares the average intra-cluster interference of the clustering scheme in this paper (SMVC) with the MAX $N$-cut [9] scheme. It can be seen

**Table 1.** Simulation parameters

| Parameters | Value |
|---|---|
| Number of V2I links | 6 |
| Number of V2V links | 25 |
| Base station height | 25 m |
| Base station antenna gain | 8 dBi |
| Vehicle height | 1.5 m |
| Vehicle antenna gain | 3 dBi |
| Noise power | −114 dBm |
| SINR threshold for V2I | 5 dB |
| V2I path loss model | 128.1+37.6*log10(d), d in km |
| V2V path loss model | LOS in WINNER+B1 |
| Shadowing distribution | Log-normal |
| Fast fading | Rayleigh fading |

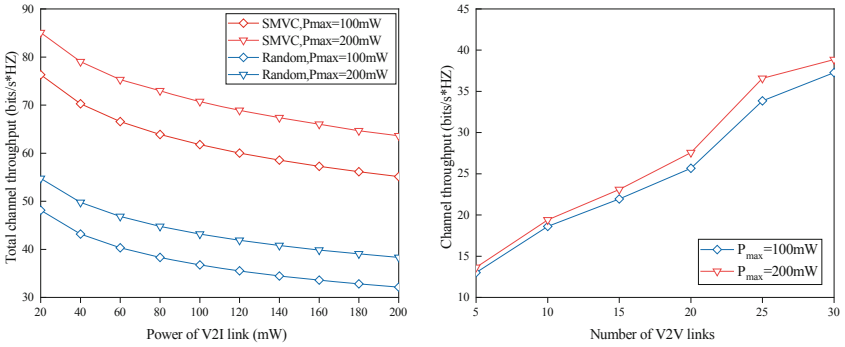



**Fig. 4.** (a) shows the effect of V2I link power on the total throughput of all channels in different cases, and (b) shows the effect of the number of V2V links in a single channel on the throughput.

that the average intra-cluster interference of our scheme is significantly lower than that of the other scheme. In addition, it can be seen that the higher the number of clusters, the lower the average intra-cluster interference.

From Fig. 4(a), it can be seen that the higher power of the V2I link, the lower total channel throughput. Besides, we show the performance of the total channel throughput with different maximum power of V2V links. It can be seen that our scheme is significantly better than the random power allocation scheme. Figure 4(b) shows the effect of the number of V2V links within a single channel on the throughput. We can see that as the number of links increases, the channel throughput shows an increasing trend.

## 6 Conclusion

This paper investigates the problem of resource allocation based on Scenario marking and vehicle clustering, where each V2I link shares a channel with multiple V2V links. Vehicles are given different attributes depending on the area in which they are located, and we classify V2V links into "altruistic links" and "ego links" according to their attributes. Considering the fairness of resource allocation, we first cluster and allocate channels to "altruistic links" to reduce intra-cluster interference. The "ego links" can choose to join channels that interfere with their own. Finally, we resort to the interior point method to solve the optimization problem of distributed power allocation, while achieving a better result.

## References

1. Liang, L., Li, G.Y., Xu, W.: Resource allocation for D2D-enabled vehicular communications. IEEE Trans. Commun. **65**(7), 3186–3197 (2017)
2. Wang, L., Ye, H., Liang, L., Li, G.Y.: Learn to compress CSI and allocate resources in vehicular networks. IEEE Trans. Commun. **68**(6), 3640–3653 (2020)
3. Ye, H., Li, G.Y., Juang, B.-H.F.: Deep reinforcement learning based resource allocation for V2V communications. IEEE Trans. Veh. Technol. **68**(4), 3163–3173 (2019)
4. He, Z., Wang, L., Ye, H., Li, G.Y., Juang, B.-H.F.: Resource allocation based on graph neural networks in vehicular communications. In: 2020 IEEE Global Communications Conference, GLOBECOM 2020, pp. 1–5. IEEE (2020)
5. Zhang, R., Cheng, X., Yao, Q., Wang, C.-X., Yang, Y., Jiao, B.: Interference graph-based resource-sharing schemes for vehicular networks. IEEE Trans. Veh. Technol. **62**(8), 4028–4039 (2013)
6. Hu, B., Chu, X.: Social-aware resource allocation for vehicle-to-everything communications underlaying cellular networks. In: 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), pp. 1–6. IEEE (2021)
7. Liang, L., Ye, H., Li, G.Y.: Spectrum sharing in vehicular networks based on multi-agent reinforcement learning. IEEE J. Sel. Areas Commun. **37**(10), 2282–2292 (2019)
8. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**(6191), 1492–1496 (2014)
9. Gyawali, S., Qian, Y., Hu, R.Q.: Resource allocation in vehicular communications using graph and deep reinforcement learning. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2019)

# Socially Acceptable Trajectory Prediction for Scene Pedestrian Gathering Area

Rongkun Ye[1], Zhiqiang Lv[1,2], Aite Zhao[1(✉)], and Jianbo Li[1]

[1] College of Computer Science and Technology, Qingdao University, Qingdao 266071, China
zhaoaite@qdu.edu.cn, lijianbo@ubinet.cn
[2] Institute of Ubiquitous Networks and Urban Computing, Qingdao University, Qingdao 266701, China

**Abstract.** Dense areas of pedestrians in complex crowded scenes tend to disrupt the proper path of the agents. The agents usually avoid gathering areas to find a reasonable pedestrian-sparse path, slow down the speed to walk, and wait for the gathering pedestrians to disperse. The accurate trajectory prediction in gathering areas is a challenging problem. This work introduces a new feature that affects trajectories to address this problem. The area gathering feature that allows agents to plan future paths based on the gathering level of pedestrians. The gathering areas as well as indicate the degree of gathering in the areas by means of a dynamic pedestrian filtering method to generate a trajectory heat map. Besides, the convolutional neural network is used to extract the corresponding area gathering feature. Furthermore, a new approach is proposed for inter-agent interactions that makes full excavation of deep interaction information and takes into account a more comprehensive interaction behavior. This work predicts trajectories by incorporating multiple factors such as area-dense features, social interactions, scene context, and individual intent. The prediction accuracy is significantly enhanced and outperforms state-of-the-art methods.

**Keywords:** Trajectory prediction · Gathering areas · Trajectory heat map · Social interactions

## 1 Introduction

Trajectory prediction predicts the path for a while in the future by studying the motion behavior of the agent. For pedestrians, the uncertainty of personal intentions, the complexity of social relationships among pedestrians, and the variability of pedestrians' surrounding environment. This makes the prediction task challenging.

In trajectory prediction, some researches [1, 2] use a social interaction pool to model the interaction between pedestrians, considering the influence of other pedestrians on the target pedestrian from the local and global aspects of the scene, respectively. However, the proposed social interaction model often identifies incorrect interaction agents. Sadeghian et al. [3] encoded the interactions between agents by a more reliable feature extraction strategy from a bird's eye view to learn the scene context. And the proposed attention


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 206–215, 2022.
https://doi.org/10.1007/978-3-031-19208-1_18

mechanism to combine scene context [4, 5] and social interaction to generate accurate interpretable social and physical feasible paths. However, the social attention mechanism is unable to memorize the social interactions of long-time pedestrians. Sun et al. [6] has built more interpretable social interaction graphs based on the relationships between pedestrians. And the deep social interaction features are extracted by graph convolutional neural network. However, the building of social interaction graphs consumes a large amount of time and resources. Besides, some work [7, 8] has used the novel thought of dividing the prediction into target points and way lines, improving the accuracy of prediction significantly.

**Fig. 1.** GA-GAN uses the trajectory heat map to highlight gathering areas, and combines the agent interaction algorithm to predict a reasonable trajectory.

This work proposes the GA-GAN method to solve the above problems. As in Fig. 1, GA-GAN considers the influence of the pedestrian gathering area on the agent's future path. Using the method of trajectory heat map to highlight gathering areas, thereby there are more attention for them in the procession of model prediction. And making full excavation of deep interaction information, GA-GAN simulates complex social interactions by pedestrians considering each agent's distance and intention. GA-GAN makes the prediction of each agent's path more reasonable and accurate by combining the information of scene context, pedestrian gathering areas, and social interactions between agents. The main contributions of this work are as follows:

- This work innovatively proposes trajectory prediction based on the fusion of gathering area features with other features and achieve excellent results.
- This work proposes a method that generates a trajectory heat map to represent the gathering areas and proposes the dynamic pedestrian filtering method (DPFM) fuzzy motion pedestrian to solve the problem of gathering area uncertainty.
- This work proposes a comprehensive and interactive approach to social interaction.
- The GA-GAN is the state-of-the-art model on the ETH/UCY dataset.

## 2 Related Work

### 2.1 Gathering Area

The study of gathering areas is generally applied in the fields of urban traffic, trajectory hotspot area discovery, and area gathering density analysis. Most of the research methods of gathering areas involve several traditional methods (K-means and some of its improved methods) [9] have detected scene hotspots by a two-stage clustering approach, in which

spatial clustering of trajectory points with spatial and temporal attributes [10–12] was applied using spatial clustering based on temporal density. Choi et al. [13] have generated a heat map based on the relational features gathered by agents in areas, and predicts the relational features of the future target location in the heat map form. In this paper, the GA-GAN generates a trajectory heat map with time dependence to represent the gathering level of an area of pedestrians at a certain period. Then combine other features to make the model predict a more accurate and reasonable path.

### 2.2 Generative Modeling

Goodfellow et al. [14] have proposed a new type of generative model, generative multi-functional networks, which is trained as a very small game between the generative and discriminative models. In Gupta et al. [2], the S-GAN has been proposed to combine GAN with trajectory prediction for the first time and proposed a social interaction model with multivariate losses to encourage GAN generative networks to extend their normal distribution and cover the space of possible paths. However, since S-GAN does not fully utilize the deep interaction information of pedestrians in the social interaction model. Many approaches have improved it by using attention mechanisms [15], adding feasibility constraints, and learning more accurate sample distribution to synthesize pedestrian interaction models and explore the influencing factors of pedestrian trajectories. In this paper, the GA-GAN makes full use of social interaction information and incorporates an attention mechanism to enhance scene interaction.

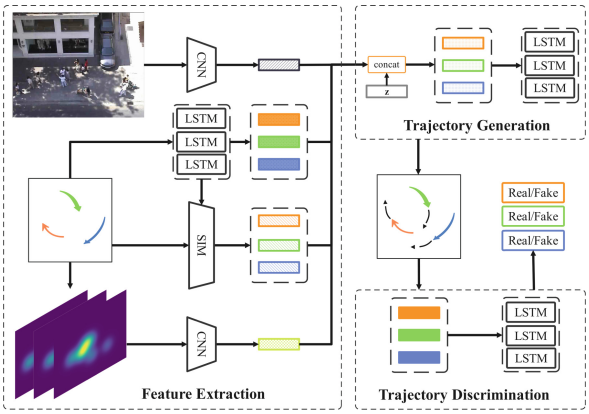



**Fig. 2.** Structure figure of GA-GAN, which consists of a feature extraction module, a trajectory generation module and a trajectory discrimination module.

## 3 Method

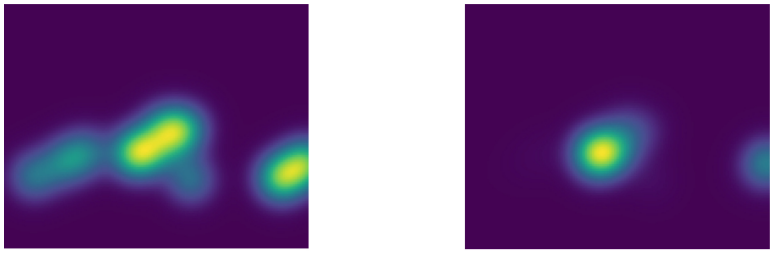
### 3.1 Problem Definition

Trajectory prediction is the process of learning the first *obs* time walker trajectories to predict the future path of the next *pred* time steps. This work defines the model input

as $\left\{X_t^i\right\} i \in N, t \in [1, obs]$, i.e., the trajectory coordinates of the $i$-th person in the current scene at the observed $t$-th time step, where $X_t^i = \left(x_t^i, y_t^i\right)$. The model output is $\left\{\widehat{Y}_t^i\right\} i \in N, t \in \left[1, pred\right]$, In addition, we define the true trajectory coordinates of the latter $pred$ time step as $\left\{\overline{Y}_t^i\right\} i \in N, t \in \left[1, pred\right]$.

### 3.2 GA-GAN

The proposed model is based on the GAN consisting of a generator and a discriminator. as shown in Fig. 2. The generator consists of a feature extraction module and a trajectory generation module, which are continuously trained to learn the path distribution of the agent and generate reasonable future trajectory samples for the agent. The discriminator consists of an LSTM-based encoder, which distinguishes whether the generated agent trajectory is feasible or not.



(a) (b)

**Fig. 3.** The (a) represents the heat map of the trajectory generated without the DPFM. The (b) represents the heat map of the trajectory generated with the DPFM.

#### Feature Extraction

*Area Gathering Feature.* We use the deep learning [16, 17] method to obtain the influence of the gathering area on the agent. Firstly, the trajectory coordinates $X_t$ of pedestrians in space need to be pre-processed, which can prevent the problem that the generated spatial heat map is inconsistent with the original scene coordinate distribution due to the inconsistent coordinate scaling of different scenes.

Next, we represent the spatially gathered areas by transforming the trajectory coordinates of pedestrians into a trajectory heat map $S_t$, as in Eq. 1. The $S_t$ is of fixed size, with width $W$ and height $H$ respectively. *GauKe* is a method we propose to generate a Gaussian kernel centered at the $k$-th pedestrian trajectory coordinate $X_t^k$ at time step $t_s$, with $r$ as the scope of influence of the agent $k$. The $n_{ped}$ is all the pedestrians in the scene at the current time steps.

$$S_t = \frac{\sum_{k=1}^{k\in\left[1,n_{ped}\right]} GauKe\left(X_t^k, r\right)}{n_{ped}} \tag{1}$$

This work proposes DPFM to solve the uncertainty of the gathering area of walking people at the current time by blurring the moving pedestrians. The DPFM reduces the impact of pedestrians with a speed greater than $v$ on the scene to highlight the areas where people are gathered, making the model easier to extract the features of the gathering areas. Here $v$ is a threshold value. If the moving speed $v_{mov}$ is above $v$, the $r$ of *GauKe* function is reduced by 2–1/*Sigmoid*($v_{mov}$-$v$) times. The effect is shown in Fig. 3. Besides, we take the trajectory heat map connection of the first $t_s$ time steps to prevent the gathering areas from changing. At last, as Eq. 2 using convolutional neural network to down-sample the trajectory heat map to extract the area gathering features.

$$A_t = CNN\left(S_{t-t_s:t};\ W_{cnn}\right) \tag{2}$$

*Individual Behavioral Feature.* The GA-GAN uses a long short-term memory (LSTM) network as an encoder to capture the temporal dependencies of the observed agent trajectory coordinates. The previous t-step trajectory coordinate of agent $i$ $X_t^i$ is then input to the $LSTM_{en}$ to obtain the behavioral features $B_t^i$, as in Eq. 3.

$$B_t^i = LSTM\left(X_{:t}^i,\ h_{en_t-1}^i;\ W_{en}\right) \tag{3}$$

*Environmental Feature.* The GA-GAN uses VGG-16 to process each frame of the video $I$ to extract the environmental features $E_t$, and uses a self-attention mechanism to emphasize areas with high impact, as in Eq. 4.

$$E_t = ATTEN\left(VGG - 16\left(I_t,\ h_t;\ W_{vgg-16}\right)\right) \tag{4}$$

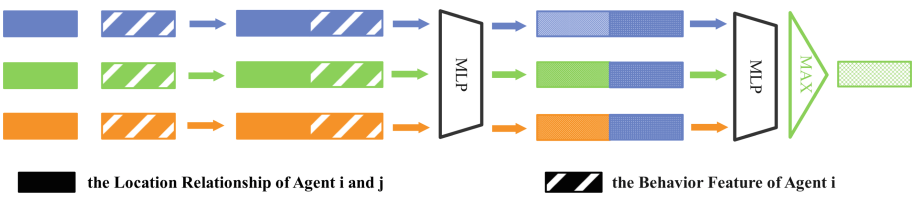



**Fig. 4.** The internal structure diagram of SIM about agent $i$

*Social Interaction Feature.* The social interaction of the agent is related to the spatial relationship between the agents in the scene and the intention of the agents. The social interaction module (SIM) of GA-GAN obtains the social interaction representation of agent $i$ based on the location relationship $L_{rel}$ between agent $i$ and other agents and the personal intention of agent $i$ to get the social interaction representation of agent $i$, as shown in Eq. 5. In SIM, the influence of other agents on agent $i$ in terms of position and behavior is inferred from the relative position of agent $i$ with other agents and their behavior, as shown in Fig. 4. Then the pedestrian with the greatest influence among the influence of other agents over agent $i$ based on the behavior of agent $i$ is selected as the interaction object of agent $i$.

$$S_t^i = SIM\left(L_{relt}^i,\ B_t^i;\ W_{SIM}\right) \tag{5}$$

**Trajectory Generation**
The trajectory generation fuses the above features and white noise vector $z$ sampled from a multivariate normal distribution to obtain a multi-source feature that can describe and constrain agents. The multi-source features are fed into the multilayer perceptron to obtain a semantic vector, which adequately represents the multi-source information. And then the vector is input into the LSTM-based decoder, which generates reasonable paths that conform to the realistic trajectory pattern and are robust by learning the relationship between the multi-source features of agent $i$ in the temporal dimension, as shown in Eq. 6.

$$\hat{Y}_{t+1}^{i} = Generation\left(\left[B_t^i,\ A_t,\ E_t,\ S_t^i,\ z_t^i\right],\ h_{gent_t^i};\ W_{gen}\right) \tag{6}$$

**Trajectory Discrimination**
The trajectory discrimination consists of an LSTM-based encoder and a MLP. The encoder estimates the time-dependent future state of the trajectory of pedestrian $i$ by learning the distribution of the trajectory of the input pedestrian $i$. The MLP discriminates the trajectory of pedestrian $i$ based on the temporal dependencies of the trajectory of pedestrian $i$ estimated by the encoder. Discriminator final output $C^i$, as shown in Eq. 7. If $C^i$ is closer to 1 then the input trajectory is more like the real trajectory, else is closer to 0 then the input trajectory is more like the faked trajectory.

$$C^i = Discriminator\left(Y^i, h_{dis}^i; W_{dis}\right) \tag{7}$$

## 4 Experiment

### 4.1 Dataset and Implementation Details

In this work, we use ETH/UCY dataset to test the interaction and capture function to the aggregation area of the model, and then evaluate the accuracy and rationality of the model. ETH/UCY dataset has a large number of rich interaction scenarios for everyone, such as gathering, following, pooling, and collision. We sample coordinate points in meters for pedestrians every 0.4s, and use Average Displacement Error (ADE) and Final Displacement Error (FDE) as evaluation metrics.

This work implement GA-GAN in PyTorch, and perform all experiments with Nvidia 3090 GPUs. The default $H$ and $W$ of trajectory heat map $S_t$ are 144 and 180 respectively, and $t$ on $S_t$ is 3. In addition, the $r$ in *GauKe* is 40, the $v$ in DPFM is 0.5. We use the Adam optimizer with default parameters and initial learning rate $1 \times 10^{-3}$.

### 4.2 Quantitative Analysis

We compare GA-GAN against several similar and SOTA baselines (S-LSTM, S-GAN, SoPhie, Y-net) which have introduced in the chap 1.

In the UCY/ETH dataset, the results of comparing GA-GAN with baselines are shown in Table 1. The effect of GA-GAN is significant in UNIV, ZARA1, and ZARA2

**Table 1.** Results of GA-GAN with baselines under the ETH/UCY dataset with a prediction time of 3.2s or 8 time steps.

| Model | S-LSTM | | S-GAN | | SoPhie | | Y-net | | Ours | |
|---|---|---|---|---|---|---|---|---|---|---|
| Metric | ADE | FDE | ADE | FDE | ADE | FDE | ADE | FDE | ADE | FDE |
| ETH | 1.09 | 2.35 | 0.81 | 1.52 | 0.70 | 1.43 | 0.28 | **0.33** | **0.23** | 0.34 |
| HOTEL | 0.79 | 1.76 | 0.72 | 1.61 | 0.76 | 1.67 | **0.10** | **0.14** | 0.17 | 0.25 |
| UNIV | 0.67 | 1.40 | 0.60 | 1.26 | 0.54 | 1.24 | 0.24 | 0.41 | **0.14** | **0.23** |
| ZARA1 | 0.47 | 1.00 | 0.34 | 0.69 | 0.30 | 0.63 | 0.17 | 0.27 | **0.14** | **0.22** |
| ZARA2 | 0.56 | 1.17 | 0.42 | 0.84 | 0.38 | 0.78 | **0.13** | 0.22 | **0.13** | **0.18** |
| AVG | 0.72 | 1.54 | 0.58 | 1.18 | 0.54 | 1.15 | 0.18 | 0.27 | **0.16** | **0.24** |

scenes with pedestrian gathering and complex interactions, which all outperform other models. Especially in UNIV, ADE, and FDE are significantly decreased compared with baselines. As UNIV has the most and densest pedestrians in all scenes, which helps GA-GAN focus more on the interactions between agents and the gathering areas, thus enabling GA-GAN to extract more important interaction features and gathering features. In addition, the environmental feature of the interaction between self-attention pedestrians and the environment extracted by GA-GAN play a guiding role in predicting the trajectory. The combination of these features can generate a more reasonable trajectory in the prediction process.

### 4.3 Qualitative Analysis

In this section, we investigate the ability of GA-GAN to model pedestrian interaction and perceive gathered areas. First two subsections investigate the impact of region gathering features extracted by GA-GAN on future trajectories. Last subsection investigates the impact of GA-GAN's improved pedestrian interaction method on future trajectories. We select ZARA with high pedestrian gathering and strong interactive behavior.

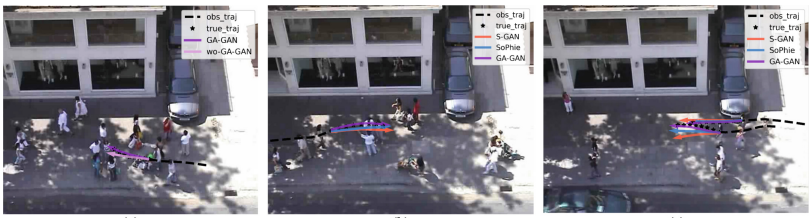



**Fig. 5.** Comparative graph of qualitative analysis results

**Gathering Versus No-Gathering**

In this subsection, we investigate the impact of the presence or absence of gathering

modules on trajectory prediction, to understand the importance of aggregation modules on GA-GAN prediction. In this scenario, pedestrians pushing baby strollers choose to bypass the gathering crowd when they encounter the gathering pedestrians in front of them. The prediction results are shown in Fig. 5 (a). It is clear that the GA-GAN identifies and perceives the gathering area in advance by the DPFM, and combines the constraints of the surrounding environment with the behavior implication of the surrounding agents, so as to focuses its prediction more on the pedestrians gathered in front. Therefore, GA-GAN adjusts the motion trajectory to avoid collision with the forward gathering pedestrians before the agent passes the gathering area. However, the GA-GAN with no gathering module (wo-GA-GAN) does not notice the forward gathering area. Therefor the gathering module of GA-GAN has a powerful function of perceiving the gathering area in advance. And then combine with multiple feature information to make the corresponding behavior of avoiding or approaching the gathering area.

**Gathering Module**
In this subsection, we choose S-GAN and SoPhie, which are based on GAN but do not include the gathering module, to compare with GA-GAN. In this scenario, the labeled agents change their walking direction to avoid pedestrians gathered in front of them when they encounter them. The prediction results are shown in Fig. 5 (b). Thanks to the inclusion of dynamic crowd gathering area feature extraction and DPFM, the GA-GAN pre-perceives the gathering area and makes avoidance behavior. However, the S-GAN, which only considers interactions related to distance and individual behavior, does not perform well with SoPhie, which fails to balance physical constraints with social interactions. Therefore, the trajectories predicted by the gathering module in complex and pedestrian-aggregated scenes are significantly enhanced in terms of both accuracy and plausibility.

**Social Interaction Module**
In this subsection, we compare SGAN and SoPhie with interaction modules against GA-GAN. In this scenario, a step of pedestrian walks slowly waiting for another pedestrian, and converges with him. The effect is shown in Fig. 5 (c). As GA-GAN uses a social interaction module with mutual influence between agents' behaviors, it can accurately measure the influence of the agent and other agents in location and behavior on this agent, and then determine the corresponding interaction behavior and the interacting pedestrians. The SoPhie measures the importance of distance and interaction by the self-attention mechanism. In this scene, the SoPhie tends to interact with the rear pedestrians but the tendency is tiny. The S-GAN selects the pedestrian in the global interaction who has the most influence on the target pedestrian in terms of distance and behavior, but not the rear converging pedestrians. Therefore, GA-GAN has greater interaction capability and easier to observe the interaction between pedestrians.

## 5 Conclusion

This paper proposes a trajectory heat map representation of the gathering areas to solve the problem of failing to model complex relationships in crowded scenes. Besides, we

propose a more interactive method for pedestrians to consider each other's intentions. This solves the problem of inconsistent interaction goals caused by both agents not considering each other. Experiments show that the GA-GAN is a state-of-the-art method.

**Acknowledgement.** National Natural Science Foundation of China under Grant No. 62106117, and Shandong Provincial Natural Science Foundation under Grant No. ZR2021QF084.

## References

1. Alahi, A., Goel, K., Ramanathan, V., et al.: Social LSTM: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 961–971. IEEE, New York, USA (2016)
2. Gupta, A., Johnson, J., Fei-Fei, L., et al.: Social GAN: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2255–2264. IEEE, Salt Lake City, USA (2018)
3. Sadeghian, A., Kosaraju, V., Sadeghian, A., et al.: Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1349–1358. IEEE, Long Beach, USA (2019)
4. Lv, Z., Li, J., Li, H., et al.: Blind travel prediction based on obstacle avoidance in indoor scene. Wirel. Commun. Mob. Comput. 1–14 (2021)
5. Jiang, B., Li, Y.: Construction of educational model for computer majors in colleges and universities. Wirel. Commun. Mob. Comput. 1–9 (2022)
6. Sun, J., Jiang, Q., Lu, C.: Recursive social behavior graph for trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 660–669. IEEE, Seattle, USA (2020)
7. Mangalam, K., An, Y., Girase, H., et al.: From goals, waypoints & paths to long term human trajectory forecasting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15233–15242. IEEE, Montreal, Canada (2021)
8. Wang, C., Wang, Y., Xu, M., et al.: Stepwise goal-driven networks for trajectory prediction. IEEE Robot. Autom. Lett. 1–11 (2022)
9. Li, F., Shi, W., Zhang, H.: A two-phase clustering approach for urban hotspot detection with spatiotemporal and network constraints. IEEE J. Select. Top. Appl. Earth Observ. Rem. Sens. **14**, 3695–3705 (2021)
10. Lv, Z., Li, J., Dong, C., et al.: DeepSTF: a deep spatial–temporal forecast model of taxi flow. Comput. J. 1–16 (2021)
11. Cheng, Z., Rashidi, T.H., Jian, S., et al.: A spatio-temporal autocorrelation model for designing a carshare system using historical heterogeneous data: policy suggestion. Trans. Res. Part C Emerg. Technol. **141**, 103758 (2022)
12. Wang, Y., Lv, Z., Zhao, A., et al.: A deep spatio-temporal meta-learning model for urban traffic revitalization index prediction in the COVID-19 pandemic. Adv. Eng. Inform. 1–17 (2022)
13. Choi, C., Dariush, B.: Looking to relations for future trajectory forecast. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 921–930. South Korea (2019)
14. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial nets. Adv. Neural. Inf. Process. Syst. **27**, 1–10 (2014)

15. Song, Y., Bisagno, N., Hassan, S.Z., et al.: Ag-gan: An attentive group-aware gan for pedestrian trajectory prediction. In: 2020 25th International Conference on Pattern Recognition, pp. 8703–8710. IEEE, Milan, Italy (2021)
16. Lv, Z., Li, J., Dong, C., et al.: Deep learning in the COVID-19 epidemic: a deep model for urban traffic revitalization index. Data Knowl. Eng. **135**, 101912 (2021)
17. Zhao, A., Wang, Y., Li, J.: Transferable self-supervised instance learning for sleep recognition. IEEE Trans. Multimedia, 1–15 (2022)

# Wi-KF: A Rehabilitation Motion Recognition in Commercial Wireless Devices

Xiaochao Dang(✉), Yanhong Bai, Daiyang Zhang, Gaoyuan Liu, and Zhanjun Hao

College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, Gansu, China

dangxc@nwnu.edu.cn, zhanjunhao@126.com

**Abstract.** Wireless sensing is increasingly used in the field of medical rehabilitation because of its advantages of low cost, non-contact and wide coverage. In the rehabilitation of patients, the recovery after upper limb injury is greatly significant. Nonstandard rehabilitation motions will cause secondary injury to the body. Therefore, how to achieve standardized rehabilitation motions at a low cost in the home environment has become an urgent problem to be solved. In order to settle it, a rehabilitation motion recognition method Wi-KF based on Wi-Fi signal is designed. First, we propose a data segmentation and counting Peak method, which can accurately segment a continuous motion into multiple single motions and lays a foundation for a feature extraction algorithm. The motion segmented by the Peak method is converted into a Doppler feature image. Then Bag of Convolutional Feature (BoCF) algorithm is used to extract features and overcomes the difference in image size. Finally, the extracted features are input into Extreme Learning Machine (ELM) algorithm for classification. The Wi-KF method has been extensively and fully verified in two real environments. The experimental results show that the average motion recognition rate of the Wi-KF method is about 94.9%. Hence the method has strong robustness. In sum, the method proposed in the paper provides a low-cost solution for standardizing the rehabilitation motions of patients.

**Keywords:** Wi-Fi · Motion recognition · Channel state information · Extreme learning machine

## 1 Introduction

Upper limb injury will directly affect the quality of life of patients. According to the medical report [1], upper limb injury involves health, including psychology, quality of life and work. Therefore, it is essential to restore upper limb function. The traditional upper limb rehabilitation training usually needs to go to the hospital with the help of professional medical equipment and rehabilitation doctors to achieve the purpose of rehabilitation training [2], or let the patient self-rehabilitation training at home, but the cost is high. With the development of science and technology, recognition of rehabilitation training motions based on wearable devices [3] and machine vision [4] has emerged. Wearable devices based on rehabilitation training motion recognition require patients to


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 216–228, 2022.
https://doi.org/10.1007/978-3-031-19208-1_19

wear multiple devices simultaneously to achieve a better recognition effect. However, it will affect the patient's experience and the high deployment and maintenance cost of sensors. Rehabilitation training motion recognition based on machine vision mainly uses optical cameras to obtain high-precision recognition results. But it is easily affected by the light intensity and is more sensitive to deployment in the bedroom. Therefore, better technology is needed to solve these problems.

With the development of wireless communication and passive sensing technology, human perception technology based on Wi-Fi signal has become a research hot spot because of its low cost, no dead corners and no contact. Previous scholars used Receive Signal Strength (RSS) to perceive human activities. However, RSS belongs to Media Access Control (MAC) layer, so it is easy to be affected by noise and is not easy to remove. On the contrary, Channel State Information (CSI) has significant advantages. It is a channel feature extracted from the Physical (PHY) Layer and a more detailed description of the channel. It can capture the multi-path changes of the signal propagation path and has strong noise resistance. Therefore, the CSI can clearly describe the impact of various human behaviors on signal propagation. Recently, the CSI has been widely used in motion detection [5], fall detection [6], gesture recognition [7], breath and heartbeat detection [8]. Literature [9] uses the combination of amplitude and phase in CSI to identify daily human activities and achieves an accuracy of 96.7%.

In order to solve these problems, we propose a rehabilitation motion recognition method Wi-KF. Our main contributions are summarized as follows.

- We use the ubiquitous commercial Wi-Fi facilities to design a rehabilitation training motion recognition system. This system can help patients with autonomous rehabilitation training at a low cost and without carrying equipment.
- To solve the spectrum diagram size difference problem, we propose introducing the BOF module into CNN-ELM to form a new BoCF-ELM model. The model can extract any size of feature images and obtain high-precision classification results.
- We analyze and evaluate the system performance of data sets in two environments. The experimental results show that the average recognition rate of rehabilitation motion is 94.9% in two natural environments. Proves the system has strong robustness and practicability.

## 2 Relates Work

### 2.1 Non-Wi-Fi Based Motion Recognition

Presently, non-Wi-Fi rehabilitation motion recognition mainly includes wearable devices and machine vision. Wearable devices analyze and recognize motion data obtained by sensors. For example, literature [3] uses human body sensors to get patients' motion information, extracts the motions' characteristics through kernel principal component analysis, and then trains and classifies them with a deep recursive neural network. Although the recognition rate of wearable sensors is high, it will cause some interference to human activities. The computer vision-based approach performs motion recognition by analyzing video streams of human movement. For example, literature [4] obtains patients' activity data through machine vision. It uses the attention mechanism to build a gated

recurrent unit network integrating three-layer temporal features to realize the classification of human rehabilitation motions. Rehabilitation training motion recognition based on machine vision will expose patients' privacy, and the complexity of video processing is high.

### 2.2 Wi-Fi Based Motion Recognition

Motion recognition based on Wi-Fi obtains the classification result by processing and analyzing the CSI data in the Wi-Fi signal. Literature [10] proposes Wi-Motion, a Wi-Fi-based human activity recognition method, constructs a classifier through the amplitude and phase information of CSI and combines the classifier's output through a combination strategy based on posterior probability. Literature [11] extracts features from fine-grained Wi-Fi channel state information and feeds them back to the deep learning model to realize action recognition at different positions. Literature [12] proposes a Wi-Fi CSI based human activity recognition approach using a deep recurrent neural network (HARNN). This method uses a weighted moving average to smooth the original CSI data. Then uses the moving variance of CSI to detect the beginning and end of activities, clusters the effective subcarriers according to the sensitivity of subcarriers, and uses SVM to classify activities.

## 3 Related Theories

### 3.1 Channel State Information

Wi-Fi perception is to realize the perception of human behaviour, object and environment in the propagation space by analyzing the wireless signal propagation channel characteristics. The propagation characteristic of this wireless signal is CSI. In other words, CSI is the compensation of various channel effects at the receiving end, such as signal reflection, refraction, diffraction and multi-path attenuation, that is, Channel Impulse Response (CIR). After Fast Fourier Transform (FFT) processing, the Channel Frequency Response (CFR) can be obtained. CFR can provide amplitude and phase information of subcarrier level through Multiple Input Multiple Output (MIMO) and Orthogonal Frequency Division Multiplexing (OFDM)technology. It can be seen from the literature [7] that CFR can be expressed as:

$$H(f,t) = \left( \sum_{i=1}^{M} \alpha_i(f,t) e^{-j2\pi f \tau_i(f,t)} \right) e^{-j(2\pi t \Delta f + \theta_N + \theta_M)} \tag{1}$$

where $\alpha_i(f,t)$ is the amplitude, $\tau_i(f,t)$ is the phase information, and $M$ is the number of subcarriers.

### 3.2 Doppler Frequency Shifts

Doppler Frequency Shifts(DFS) are caused by the relative position change of the target, receiver and transmitter. In the non-contact sensing environment, the transmitter and

receiver are statically deployed, and the actions of the human body change the path of Wi-Fi signal transmission. When the human body acts between the transmitting end and the receiving end, the peaks and troughs of the reflected electromagnetic wave signal reach the receiver at a faster speed. Still, when there is no action, it is relatively stable. Generally speaking, the DFS can be obtained by the time-frequency analysis for CSI.

$$H(\tau) \approx h(\tau)\mathrm{e}^{-\mathrm{j}2\pi\left(\overset{\Delta}{t}f+\overset{\Delta}{f}t\right)} + \sum_{n\in D} a_i B\left(f_{D_n}(t)\right) \tag{2}$$

where $B\left(f_{D_n}(t)\right)$ is the window function of intercepting the CSI action signal segment, $h(\tau)\mathrm{e}^{-\mathrm{j}2\pi\left(\overset{\Delta}{t}f+\overset{\Delta}{f}t\right)}$ is the phase offset due to the lack of synchronization between Wi-Fi network cards, resulting in the unknown phase offset in the original CSI. $2\pi\left(\overset{\Delta}{t}f+\overset{\Delta}{f}t\right)$ is the phase offset, carrier frequency and time offset.

## 4 Systems Design

### 4.1 System Overview

The Wi-KF method mainly includes data acquisition, processing, feature extraction and classification. The specific process is shown in Fig. 1:

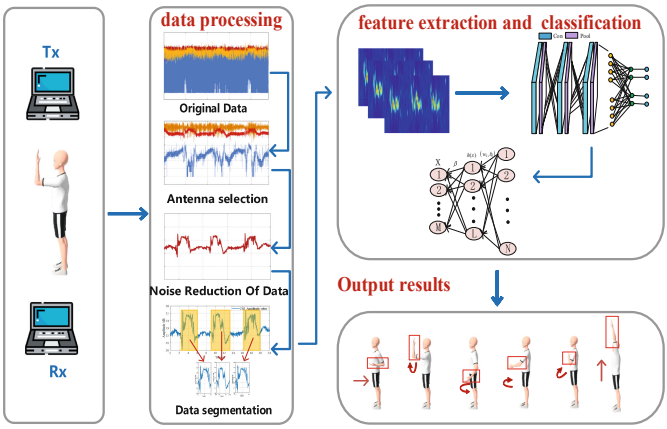



**Fig. 1.** Wi-KF flow chart.

The data processing flow in Fig. 1 is: select the antenna and then use Butterworth low-pass filter and Symlet 8 (Sym8) wavelet to process the abnormal data of the original data. Then the processed data are segmented and counted, and the single-segmented motion is transformed into a spectrum. Secondly, use BoCF to extract spectral features. Finally, elm trains and classifies the extracted features.

### 4.2 Data Acquisition and Processing

The six selected rehabilitation movements are shown in Fig. 2, namely "posterior flexion", "anterior flexion", "external rotation", "internal rotation", "internal retraction" and "superior extension" are the six common movements used to assess the method.

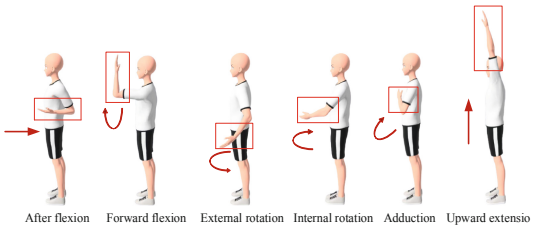



**Fig. 2.** Six rehabilitation motions.

Traditional CSI motion recognition selects the channel characteristics of single antenna as the acquisition source of perceptual data [13].This method will lose more motion feature data. We use the method of variance in reference [14] to select the antenna, because variance helps to feedback the impact of motion change on CSI, it can better respond to dynamic response. Therefore, we choose the antenna with the largest amplitude and the smallest variance of CSI and the antenna with the smallest amplitude and the largest variance.

Due to the substantial environmental noise when collecting data, denoising is needed. We select Butterworth low-pass filter [15] and Sym8 wavelet [16] to process the noise of the collected original CSI data. The data processing process is shown in Fig. 3:

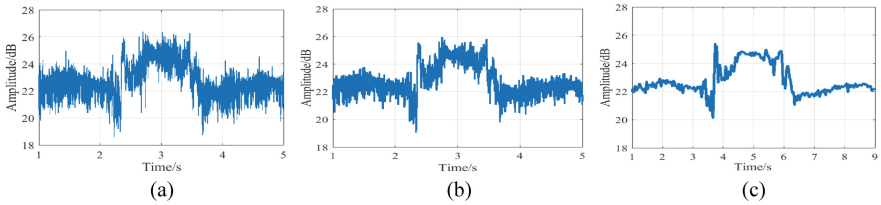
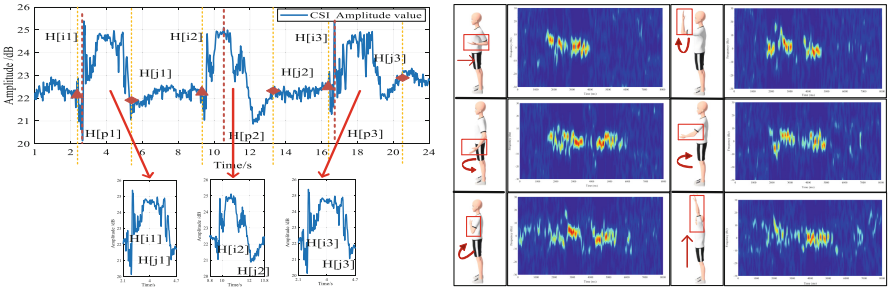



**Fig. 3.** Processing process of combined filtering method. (a) Original data. (b) Butterworth filtered subcarrier. (c) Wavelet filtered subcarrier.

As can be seen from Fig. 3 (a), the original subcarrier data has a lot of noise, which may reduce the accuracy of motion recognition. Figure 3 (b) is the result of the Butterworth filter filtering out the high-frequency noise interference in the original data. From the figure, it can be seen that the noise in the original data is basically removed. Figure 3 (c) the image after Sym8 filtering based on Butterworth low-pass filter. At this time, the data becomes smoother than before and retains the characteristics of the signal to the greatest extent.

Wi-KF method needs to segment and count continuous actions. Many experiments show that the CSI data flow is stable when the user does not perform any action. In

contrast, when the user acts, the fluctuation range of CSI data flow is greater than the steady-state amplitude change. Therefore, according to the above rules, we propose a peak algorithm. A pause of 1 to 2 s is required between each action. Its idea is: to scan from the left side of the data flow. If the data starts to move slowly from the static environment to the dynamic environment and then to the stable environment, action is counted at this time. By analogy, count the number of continuous actions. The starting point of the action is where the amplitude of each action continues to rise. The endpoint of the action is the starting point when the amplitude continues to drop to a steady state. Use this law to find each motion's starting point and ending point. The results of Peak algorithm data segmentation are shown in Fig. 4 (a):




(a) (b)

**Fig. 4.** Data processing.(a) Data segmentation. (b) Spectrum diagram of each motion.

It can be seen from the segmentation results. The algorithm can segment the start and the end of the motion. In the figure, the triangle represents the beginning of the motion, and the diamond represents the end. $H[i]$ means the amplitude value of the motion start point, and $H[j]$ represents the amplitude value of the motion endpoint. $H[p]$ represents the amplitude value of the motion trough.

We convert the CSI data into a spectrum diagram for time-frequency analysis to further extract the motion information. We take the single-segmented action as the input of a short-time Fourier transform (STFT) to extract the Doppler shift feature. Finally, the spectrum diagram of each motion is shown in Fig. 4 (b):

It can be seen from the spectrum diagrams of the above six motions that the spectrum diagrams of each motion are different, indicating that different motions have other effects on the propagation different of the Wi-Fi signal.

### 4.3 BoCF Feature Extraction

We propose a new model BoCF-ELM, which combines CNN-ELM and Bag Of Feature (BOF). The model solves the problem of sample size differences in images. The structure of BoCF-ELM is shown in Fig. 5:

In our model, the spectrum diagram of CSI is fed to the convolution layer as input. Then, convolution between the previous layer and a series of learnable kernel filters is performed. Local features are obtained from the input spectrum diagram. Then add the

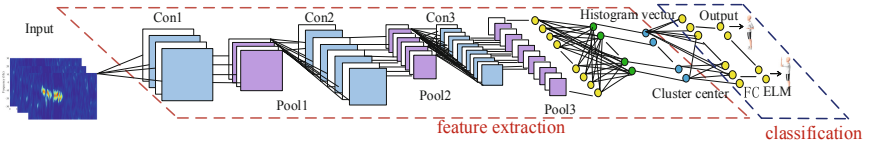



**Fig. 5.** BoCF-ELM structure diagram.

offset to the output of the convolution. The input and kernel filter of the previous layer can be expressed as:

$$z_j^l = \sum_{l=1}^{M^{l-1}} \left( H_i^l \otimes k_{i,j}^{l-1} \right) + b_j^l \tag{3}$$

where $H_i^l$ represents the input of the $l$ nd layer and $z_j^l$ represents the output of the current layer. $k_{i,j}^{l-1}$ is defined as a weighted kernel filter from the $i$ th neuron in layer $l-1$ to the $j$ th neuron in layer $l$. $\otimes$ is convolution operation, and the size of our convolution kernel is set to $5 \times 5$ and $b_j^l$ represent the bias of the $j$ th neuron in layer $l$. $M^{l-1}$ indicates the number of kernel filters in layer $l-1$. Then, we use the $ReLU = \max(0x)$ activation function on the output of the convolution layer to solve the disappearance or explosion gradient problem.

After the average pooling layer operation, obtain the output of layer $l$:

$$y_j^l = \text{ave\_ pool}\left( H_j^l \right) \tag{4}$$

The purpose of averaging pooling is to detect more useful features when down sampling the input element map. We set the pooling kernel size $2 \times 2$ to and reconstruct the output of the last pooling layer as the input of the next layer, $H_j^{l+1} = y_j^l$, which is a matrix. The processes of other convolution and average pool layers are the same as those of each layer above, and the parameter settings are the same.

We transfer the features extracted from the last average pooling layer to the BOF quantization part. After extracting the features of the $j$ th spectrum diagram, the set $x_{ji} = (i = 1 \cdots M_{\text{j}})$ of $M_j$ feature vectors is obtained. We can obtain a histogram vector with a fixed dimension by quantifying the feature vector into our predetermined clustering centre. The size of the histogram is independent of the dimension of the input image. Therefore, the model can extract image features of any size.

To make the output of the BOF module bounded, we adopt the $L^1$ normalization method. Therefore, the output B of the $n$ th neuron in the cluster centre is defined as:

$$[\phi(x)]_n = \frac{\exp(-||(x - v_n)||2/\sigma_n)}{\sum_{m=1}^{N_n} \exp(-||(x - v_m)||_2/\sigma m)} \tag{5}$$

In formula (5), $x$ is the feature vector extracted by CNN. $V_n$ represents the $n$ th neuron in the cluster center. The neuron in each cluster center has a parameter $\sigma$. It is used to

adjust the degree of quantification. $\sigma_n$ is the number of neurons in the cluster center. The number of neurons in the cluster center in the model is allowed to take different values.

The encoded histogram vector is defined as:

$$s_j = \frac{1}{N}\sum_{i=1}^{N_j}\phi(x_{ji}) \tag{6}$$

$x_{ji}$ represents the $i$th feature vector of the $j$th image. $N_j$represents all feature vectors of the $j$th image. The obtained histogram vector $s_j$ can be transmitted to ELM classifier for classification $x_{ji}$ represents the $i$th feature vector of the $j$th image $N_j$represents all feature vectors of the $j$th image. The obtained histogram vector $s_j$ can be transmitted to ELM classifier for classification.

### 4.4 ELM Classification Algorithm

After the full connection layer, ELM is used to classify the one-dimensional vector of feature graph transformation. ELM was first proposed by Huang et al. [17], and it is a new fast-learning algorithm. It does not need to adjust the parameters in the training process but only needs to set the number of neurons in the hidden layer. Compared with the traditional classification algorithm, this has the advantages of fast learning speed, strong generalization ability and fewer adjustment parameters.

The general steps of the ELM algorithm can be described as follows:

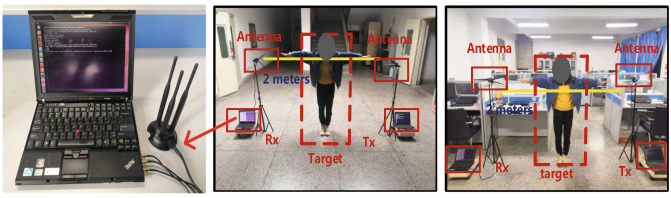
**Step 1:** Input given training sample $\{(\boldsymbol{x}_i, \boldsymbol{t}_i)\}_{i=1}^{N} \subset \mathbb{R}^n \times \mathbb{R}^m$, hidden layer output. Function $G(a, b, x)$, number of hidden layer nodes $L$.
**Step 2:** Random generation of the hidden layer node parameter $(\boldsymbol{a}_i, b_i), i = 1, \cdots, L$.
**Step 3:** Compute the hidden layer matrix (ensuring full rank in column H) and the network optimal weights $\beta$:$\beta$=$H^{\dagger}T$.
**Step 4:** Output ELM value $f(\boldsymbol{x})$ : $f(\boldsymbol{x}) = h(\boldsymbol{x})\hat{\beta} = h(\boldsymbol{x})\boldsymbol{H}^{\dagger}\boldsymbol{T}$.

## 5 Set-Up of the Experimental and Analysis of the Experimental

### 5.1 Experimental Setup

In order to verify the feasibility of Wi-KF method in the actual scene. We adopt the IWL 5300 NIC scheme based on the IEEE 802.10n protocol. The specific configuration is two laptops equipped with IWL 5300 NIC and CSI Tool, one laptop as the transmitter (Tx) and the other as the receiver (Rx). Tx has one antenna, and Rx has three antennas. The antenna contacts at the transmitting and receiving end are connected with an external antenna of 1.5 m. The experiment selects two real scenes, as shown in Fig. 6: Hall and office.

We selected 20 experimenters aged 23–70 to collect CSI data. All motions are completed in the designated area. The distance between transceivers is 2 m, which is very short because a high signal-to-noise ratio can be obtained. Each motion was collected 20 times from each volunteer, and 7200 samples were collected for all scenes. These




(a) (b) (c)

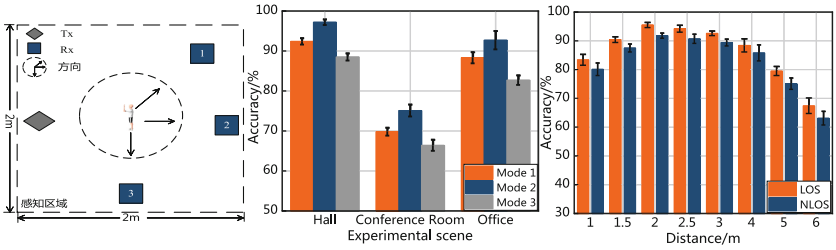
**Fig. 6.** Experimental equipment and environment. (a) Experimental equipment (b) An empty hall. (c) Simple meeting room.

samples are used to establish the data set of our motions. The training and test samples are in the ratio of 4:1. To make the experimental results more concise, we abbreviate after flexion, forward flexion, external rotation, internal rotation, addition and upward extension as AF, FF, ER, IR, AD and UE respectively.

## 5.2 Experimental Analysis

### 5.2.1 Influence of Equipment Location and Environment

In the experimental scenario, verify our method's performance in the new environment and the impact of device location. We have designed three equipment positions, as shown in Fig. 7 (a). The specific design of the experiment is standby distance: the position of the experimenter and the position of the transmitting end are fixed, and only the position of the receiving end is changed. Experiments were conducted in lobby, office, and new meeting room environments, and the results are shown in Fig. 7 (b). To find the best equipment distance, we have carried out experiments in Line of Sight (LOS) and Non-Line of Sight (NLOS) environments, respectively, and the comparison results are shown in Fig. 7 (c):




(a) (b) (c)

**Fig. 7.** Impact of equipment location and environment. (a) Equipment location. (b) Impact of equipment location. (c) Influence of device distance.

As can be seen from the experimental results in Fig. 7 (b), when the experimenter and the equipment are in a straight line, the recognition rate is the highest. On the contrary, position 1 has the lowest accuracy because the distance between devices is closer and the distance between devices and people is farther, so the recognition rate decreases. It

can also be seen that the overall recognition accuracy of the hall is higher than that of the office environment due to the influence of the multi-path effect. The accuracy of the new environment is the lowest because Wi-Fi signals are greatly affected by environmental changes, Resulting in low accuracy. As can be seen from Fig. 7 (c), when the distance between the transmitting end and the receiving end is about 2 m, whether it is LOS or NLOS, its recognition rate is the highest because the signal propagation distance is short and the signal attenuation is small. The accuracy decreases when the equipment distance is 5 m or more because the signal propagation distance is long and the signal attenuation increases.

### 5.2.2 The Impact of User Diversity

To verify the impact of different experimenters making the same motion, the same experimenter doing different motions and users of different ages making the same motion on the recognition rate. We selected four experimental personnel, two men and two women, to experiment in an open hall environment. Four experimenters were asked to perform flexion rehabilitation. The experimental results are shown in Fig. 8 (a). Then ask one of the experimenters to do all the actions we proposed. The experimental results are shown in Fig. 8 (b). We asked five people of different ages to do up and out in the above environment to verify the impact of users of different ages on the Wi-KF method. The results are shown in Fig. 8 (c):

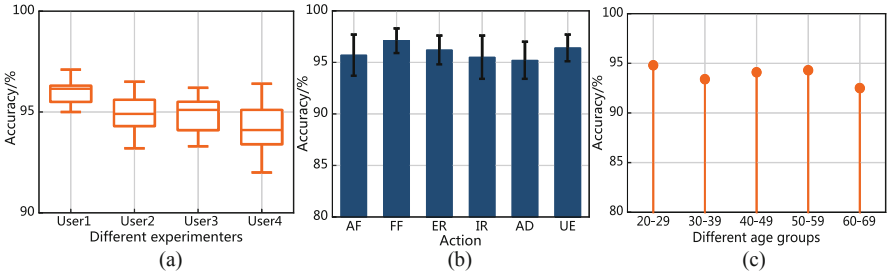



**Fig. 8.** User diversity analysis. (a) Different experimenters do the same motion. (b) The same experimenter do different motions. (c) Experimenters of different ages do the same motion.

As shown in Fig. 8 (a), the motion recognition rate of the four experimenters remained above 91%. The results show that the change of experimental personnel will not cause significant fluctuation in the experimental results. The highest recognition accuracy of experimenter 1 is 96.1% because he practised forward bending for a long time and performed standard movements. In addition, the fourth experimenter did not receive relevant training and was not good at forwarding bending, which affected the accuracy of motion recognition. As shown in Fig. 8 (b), The recognition rate of each action has reached more than 90%, indicating that different motions of the same person have little impact on the experimental results. As shown in Fig. 8 (c), the user recognition rate of the five age groups is the same, and the accuracy rate is more than 92.9%. From the experimental results, we can see that our method is suitable for all users.

### 5.2.3 Comparison of Algorithms

To verify the performance of our feature extraction algorithm, we compare them with. Histogram of Oriented Gradient (HOG), Scale Invariant Feature Transform (SIFT), and Graph Convolutional Network (GCN) features extraction algorithms. The results are shown in Fig. 9 (a). In order to verify the advantages of our hybrid model classification algorithm. We will compare it with CNN's Softmax, combined SVM and Long and Short-term Memory (LSTM) classification algorithm. The experiment was carried out in the real environment we deployed. The final comparison result is shown in Fig. 9 (b) because the number of hidden nodes in our ELM classification algorithm is an important parameter. To explore the optimal number of nodes, we selected different numbers of hidden layer neurons for comparative experiments in the laboratory environment. As shown in Fig. 9 (c).

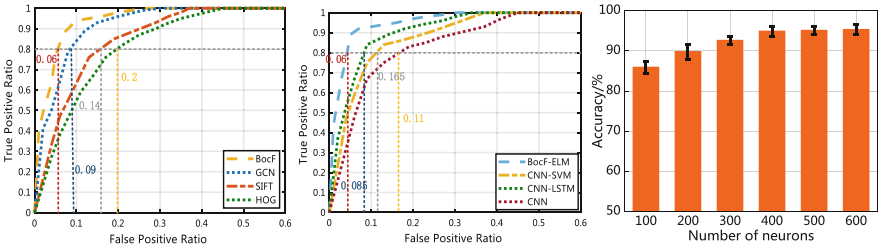



**Fig. 9.** Comparison of algorithms. (a) Comparison of feature extraction algorithms. (b) Comparison of classification algorithms. (c) Comparison of neuron number.

From the experimental results in Fig. 9 (a), the accuracy of the traditional image feature extraction algorithm is lower than that of the neural network feature extraction algorithm. This is because a neural network can automatically extract appropriate features according to the application scenario. As can be seen from the experimental results in Fig. 9 (b), when the True Positives Rate (TPR) of the classification algorithm of the BoCF-ELM model reaches 0.8, the False Positives Rate (FPR) is only 0.06, which performs best. When the TPR of CNN-LSTM is 0.8, the FPR is 0.085, slightly lower than that of BoCF-ELM. When the TPR of CNN-SVM is 0.8, the FPR is 0.10, followed by the performance. When the TPR of the CNN method is 0.8, the FPR is 0.155, and the version is the worst. Based on the above, the BoCF-ELM model has the best classification performance. This is because the elm classifier only needs to set the number of hidden layer nodes of the network. As seen from Fig. 9 (c), as the number of hidden nodes increases, the recognition accuracy gradually becomes stable. Since more hidden nodes will lead to longer training time, we set the number of hidden layer neurons to 400.

### 5.2.4 Comparison of Different Models

To verify the performance of our model, we used Wi-KF to compare with the existing Wi-Motion [10], WiAct [11], and HuAc [12] motion recognition models. Experiments

are conducted in our deployed conference room environment.Under the same conditions, we used various existing and proposed methods to recognize rehabilitation actions. The accuracy of the final comparison is shown in Table 1.

**Table 1.** Comparative results of different methods

| Method | AF | FF | ER | IR | AD | UE |
|---|---|---|---|---|---|---|
| Wi-Motion | 89.7 | 91.5 | 89.2 | 89.5 | 88.4 | 90.9 |
| WiAct | 87.6 | 89.1 | 88.6 | 89.6 | 86.9 | 89.9 |
| HuAc | 87.5 | 87.1 | 86.7 | 88.3 | 85.4 | 88.1 |
| Wi-KF | 90.4 | 94.6 | 92.1 | 92.7 | 91.5 | 95.6 |

According to Table 1, among the four recognition methods, the recognition accuracy of Wi-KF is higher than that of the other three methods. The recognition accuracy of WiAct and Wi-Motion is below 90%. Wi-KF has excellent performance. Based on the above, Wi-KF applies to the rehabilitation action recognition of people in most family environments and can provide a more accurate recognition rate and excellent robustness.

## 6 Results

This paper presents a rehabilitation motion recognition method based on CSI. Firstly, the antenna is selected by the variance method, and a combined filtering algorithm removes the noise in the original data. For the segmentation and frequency counting of continuous motions, we propose the Peak algorithm. Then the single-segmented signal is converted into a spectrum diagram. Finally, BoCF is used to extract the features of the spectrum diagram, and then the features are input into ELM for classification. The effects of different factors on the accuracy of motion recognition are analyzed in the experiment. The performance of Wi-KF in two environments is evaluated. The experimental results show that the average accuracy of this method is 94.9%. Therefore, Wi-KF can become a rehabilitation exercise training program. In addition, we will improve the recognition accuracy in future work and consider further research on multi-person recognition.

## References

1. Anderson, K.D.: Targeting recovery: priorities of the spinal cord-injured population. J. Neurotrauma **21**(10), 1371–1383 (2004)
2. Ruiz-Fernandez, D., Marín-Alonso, O., Soriano-Paya, A., García-Pérez, J.D.: eFisioTrack: a telerehabilitation environment based on motion recognition using accelerometry. Sci. World J. **2014**, 1–11 (2014)
3. Uddin, M., Hassan, M.M., Alsanad, A., Savaglio, C.: A body sensor data fusion and deep recurrent neural network-based behavior recognition approach for robust healthcare. Inf. Fusion **55**, 105–115 (2020)

4. Solongontuya, B., Cheoi, K.J., Kim, M.-H.: Novel side pose classification model of stretching gestures using three-layer LSTM. J. Supercomput. **77**(9), 10424–10440 (2021). https://doi.org/10.1007/s11227-021-03684-w
5. Bokhari, S.M., Sohaib, S., Khan, A.R., Shafi, M., Khan, A.R.: DGRU based human activity recognition using channel state information. Measurement, **167**, 108245 (2021)
6. Wang, H., Zhang, D., Wang, Y., Ma, J., Wang, Y., Li, S.: RT-Fall: a real-time and contactless fall detection system with commodity WiFi devices. IEEE Trans. Mob. Comput. **16**(2), 511–526 (2017)
7. Guo, Z., Xiao, F., Sheng, B., Fei, H., Yu, S.: WiReader: adaptive air handwriting recognition based on commercial WiFi signal. IEEE Internet Things J. **7**(10), 10483–10494 (2020)
8. Abdelnasser, H., Harras, K.A., Youssef, M.: UbiBreathe: a ubiquitous non-invasive WiFi-based breathing estimator. In: Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hangzhou China, pp. 277–286 (2015)
9. Duan, P., Li, H., Zhang, B., Cao, Y., Wang, E.: APFNet: amplitude-phase fusion network for CSI-based action recognition. Mob. Netw. Appl. **26**(5), 2024–2034 (2021). https://doi.org/10.1007/s11036-021-01734-4
10. Muaaz, M., Chelli, A., Gerdes, M.W., Pätzold, M.: Wi-Sense: a passive human activity recognition system using Wi-Fi and convolutional neural network and its integration in health information systems. Ann. Telecommun. **77**(3), 163-175 (2021)
11. Chang, J.-Y., Lee, K.-Y., Lin, K. C.-J., Hsu, W.: WiFi action recognition via vision-based methods. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, pp. 2782–2786 (2016)
12. Hao, Z., Duan, Y., Dang, X., Zhang, T.: CSI-HC: a WiFi-based indoor complex human motion recognition method. Mob. Inf. Syst. **2020**, 1–20 (2020)
13. Wang, J., et al.: A survey on CSI-based human behavior recognition in through-the-wall scenario. IEEE Access **7**, 78772–78793 (2019)
14. Qian, K., Wu, C., Zhou, Z., Zheng, Y., Yang, Z., Liu, Y.: Inferring motion direction using commodity Wi-Fi for interactive exergames. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver Colorado USA, pp. 1961–1972 (2017)
15. Weibo, L., Peiming, L., Huazhong, W.: Design and realization of the butterworth filter. In: 2009 SEG Annual Meeting. OnePetro (2009)
16. Gomes, D.P.S., Ozansoy, C., Ulhaq, A.: High-sensitivity vegetation high-impedance fault detection based on signal's high-frequency contents. IEEE Trans. Power Deliv. **33**(3), 1398–1407 (2018)
17. Huang, G.-B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **44**(1), 103–115 (2015)

# Security and Privacy

# An Effective Insider Threat Detection Apporoach Based on BPNN

Xiaoling Tao[1,2], Runrong Liu[1,2(✉)], Lianyou Fu[1,2], Qiqi Qiu[1,2], Yuelin Yu[1,2], and Haijing Zhang[3]

[1] Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
1725970716@qq.com

[2] Guangxi Cooperative Innovation Center of Cloud Computing and Big Data, Guilin University of Electronic Technology, Guilin 541004, China
txl@guet.edu.cn

[3] Department of Computer Science, Georgia State University, Atlanta, GA 30302, USA

**Abstract.** With the increasing number of insider threat incidents, insider threat has become one of the most serious network security problems. Currently, the large volume of user data generated by various network systems and devices is difficult to analyze, and it is very difficult to detect abnormal user behavior among them. Meanwhile, existing insider threat detection methods cannot fully learn the important features of user data, resulting in a high false alarm rate and low accuracy. To solve these problems, we propose a novel insider threat method based on variational auto-encoder (VAE) and back propagation neural network (BPNN) in the paper. Initially, we use the generative model VAE to construct the normal user behavior model, and obtain the effective feature representation of user behavior. Then, we use the BPNN algorithm to detect abnormal user behavior from a large number of user activity logs. Finally, we conduct experiments to verify the detection performance of the proposed method. Experimental results indicate that the proposed detection method can achieve high accuracy and precision.

**Keywords:** Insider threat detection · BPNN · VAE · Abnormal behavior

## 1 Introduction

Since Edward Snowden stole and exposed approximately 1.7 million secret documents from the National Security Agency (NSA) [1], the issue of cybersecurity has been paid more and more attention. Organizations make great efforts to address network security problems by using external protection tools, such as intrusion detection system (IDS), firewall, and antivirus software. However, these external defenses of systems cannot protect against the insider threat. Insider threat refers to "the authorized internal user who initiates malicious operations


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 231–243, 2022.
https://doi.org/10.1007/978-3-031-19208-1_20

to cause adverse effects on organization's information systems" [2]. By exploiting granted permissions and system vulnerabilities, a malicious insider (MI) or masquerader can bypass external security defenses to conduct intellectual property theft or information technology sabotage [3].

Meanwhile, frequent internal threat incidents can cause more severe damage to organizations than external threat. The Cost of Insider Threats Report from IBM Security [4] described that the global average cost of an insider threat was 11.45 million dollars. The frequency of insider incidents has tripled since 2016 from one to 3.2 per organization, and 204 organizations experienced a total of 4,716 insider incidents over the past 12 months. Currently, insider threat is one of the most serious threats to breaching cybersecurity defenses. Therefore, it is crucial to develop an effective technology to detect malicious insiders.

However, many challenges need to be addressed for insider threat detection. Firstly, in the era of big data, the explosion of user information hinders the detection of insider threat. Because it is difficult to perform data mining and analysis on large-scale and high-dimensional information. Secondly, insider attacks exhibit varying behavioral characteristics in different scenarios. At the same time, insiders can take steps to disguise anomalous behavior from being identified. In such cases, the difficulty of detecting insider threat is further increased. Hence, there is a deficiency of techniques to effectively distinguish abnormal user characteristics from massive normal user data.

Existing insider threat detection methods cannot fully learn the important features of user behavior, resulting in a high false alarm rate and false negative rate. Machine learning-based methods are difficult to deal with large-scale and high-dimensional user behavioral data, which reduces the accuracy of detection. Deep learning-based methods have high detection accuracy, but the computational overhead is great. Therefore, it is essential and urgent to design a novel method to detect insider threat effectively and accurately.

### 1.1 Contributions

The main contributions of this paper are summarized as follows:

- Many existing detection methods cannot efficiently capture important features of user behaviors, causing more false alarms. Thus, we use VAE to obtain the refined normal user behavioral representation after reducing the dimensionality of the input data.
- The large-scale and high-dimensional user behavioral data in the complex network results in low detection efficiency and poor detection performance. Thus, we use BPNN to process a large amount of behavioral data, and detect whether a user behavior record is abnormal.
- We implement the detection method based on VAE and BPNN, and then conduct the performance evaluations. The experimental results demonstrate that our approach can attain high precision.

### 1.2 Organization

The reminder of this paper is organized as follows. In Sect. 2, we review literatures related to insider threat detection. Then, we introduce our insider threat detection method based on VAE and BPNN in Sect. 3. Experiment details are described in Sect. 4. Section 5 summarizes the paper.

## 2 Related Work

Due to the detriment of insider threat, many security researchers have conducted in-depth research on insider threat in recent years. Generally speaking, the existing insider threat detection methods can be summarized into three categories: statistic-based detection method [5], machine learning-based detection method [6], and deep learning-based detection method [7].

In the early years, statistical methods were commonly used for insider threat detection. Raveendran et al. [8] described an insider detection method based on user profiles. The user profile is created by using the event co-occurrence matrix (ECM) and its covariance matrix. Taylor et al. [9] adopted the linguistic inquiry and word count (LIWC) technique to analyze the language in emails to detect abnormal insider behavior. The minimum descriptive length (MDL) technique was used to detect malicious activities in social networks, businesses and various cybercrime domains [10]. Seo et al. [11] proposed a quantitative insider threat analysis method based on the analytic hierarchy process (AHP). This method evaluated the highly probable characteristics of relevant insider threat scenarios. On the one hand, statistic-based methods can be a simple way to detect insider threat. On the other hand, the judge rules of these methods depend on subjective factors, and their detection accuracy needs to be improved.

Machine learning techniques can improve the performance of detection algorithms through self-learning. They can make up for the shortcomings of traditional statistics-based methods. Qiu et al. [12] combined sequence alignment (SA) with continuous hidden Markov model (HMM) to predict masquerader attacks. Ye et al. [13] provided a double-layer HMM framework to model user behavior, and the model was accurate in distinguishing malicious users from other users. Wall et al. [14] proposed an effective insider threat detection method based on the Bayesian network. They enhanced the efficiency of the network by providing algorithms that reduced the processing time from exponential to linear. To perform low-cost detection and adapt to user behavior changes, Chen et al. [15] applied the support vector machine (SVM) to masquerader attack detection. Le et al. [16] proposed a user-centered insider threat detection system. They used a machine learning technique to analyze malicious insider behavior on multiple data granularity levels. Although machine learning-based methods can detect insider threat precisely, their insufficient generalization ability may cause some problems. However, they cannot apply to circumstances with massive high-dimensional data.

Deep learning methods can further improve the detection accuracy and generalization ability in the task. Liu et al. [17] proposed an insider threat detection method based on deep auto-encoder (DAE), and viewed the reconstruction error of the deep autoencoder as an anomaly score to identify insider threat. Hu et al. [18] proposed a user authentication method based on the convolutional neural network (CNN), analyzing mouse biological behavior characteristics to detect anomalies. Literature [19] proposed an insider threat detection method based on the recurrent neural network (RNN), which was used to model user activities from sequential data. Meng et al. [20] proposed an attribute classification detection method based on long-short-term memory (LSTM) network. This method integrated event aggregators, feature extractors, attribute classifiers, and anomaly calculators into an end-to-end detection framework to achieve a high detection rate. Jiang et al. [21] proposed a consistent insider detection model based on graph convolutional network (GNN). This model characterized the attributes of entities and structural information between entities as a graph, and then discovered malicious behavior based on the features of the graph. Deep learning-based methods can process massive high-dimensional data with high detection accuracy, but their computational overhead is great.

The above insider threat methods both achieve very good results, but there is still a lot of room for improvement. Existing insider threat detection methods cannot efficiently capture the latent distribution of user behavioral data, which causes an increase of false positives and false negatives. Some of them cannot apply to circumstances with massive high-dimensional data. Hence, we need to adopt a technique with stronger ability to capture the difference of characteristics between normal and malicious user behavior. To deal with large-scale and high-dimensional user behavior data, it is necessary to use deep learning techniques to improve the efficiency of insider threat detection.

## 3 Our Approach

In this section, we describe the system structure and provide a detailed description of the VAE-BPNN insider threat detection method.

### 3.1 System Structure

Nowadays, the large volume of redundant user behavioral data affects the efficiency and accuracy of the detection method. Meanwhile, existing insider threat detection methods cannot efficiently capture important features of user behaviors, resulting in a very high false alarm rate and false negative rate. To solve the above issues, we propose an insider threat detection method based on VAE and BPNN. The architecture diagram of the proposed method is shown in Fig. 1.

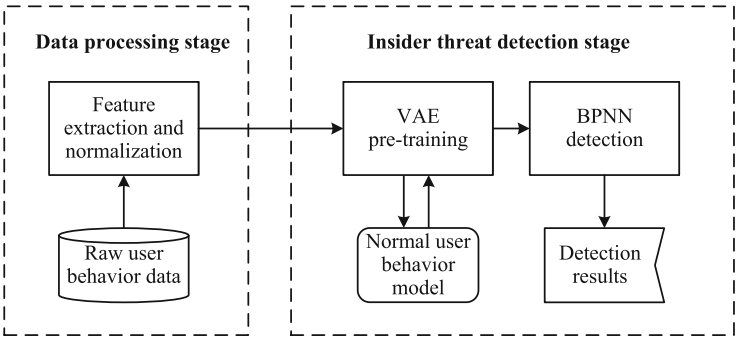



**Fig. 1.** The architecture diagram of the VAE-BPNN method.

Since the insiders' behavioral data in companys or other organizations are collected by different devices and network systems, the data is chaotic without a unified structure. Therefore, data processing is required first. In short, we should clean the original data and fill in its missing values. And the needed features for the research should be normalized, which is necessary for detection. In this way, we have completed the process of data processing.

Next, the insider threat detection stage is the core part of the VAE-BPNN architecture. In this stage, we combine VAE and BPNN to implement insider threat detection, which are also the core part of the architecture. In the beginning, we utilize the generation capability of VAE to accurately capture user behavior characteristics, thus obtaining a compressed vector. Using the compressed vector as input, we train the BPNN classification model later. Ultimately, we can get the detection result - whether the user's behavior data is malicious or normal. The implementation details of the insider threat detection method are described in the next subsection.

### 3.2 Insider Threat Detection Method Based on VAE and BPNN

In this part, we use VAE to obtain the low-dimensional representation of user behavior, and then use BPNN to detect abnormal user behavior.

**VAE Pre-training.** VAE [22] is a self-supervised neural network that learns how to encode the input into lower dimensions, then decode and reconstruct the data again to be as close to the input as possible. The VAE architecture diagram is shown in Fig. 2.

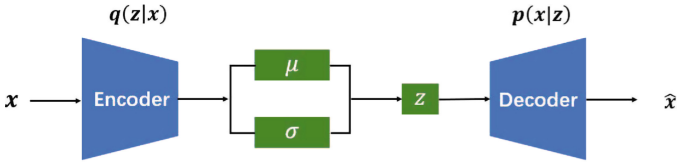



**Fig. 2.** Structure of VAE.

Here, we explain the basic training process of the VAE network, about how to construct the normal user behavior model and obtain the compressed feature representation.

Assume that the input data is $x$, and the output data is $\widehat{x}$. First, we use the probabilistic encoder $q(z|x)$ to encode the input data into a low-dimensional representation $z$. To accurately simulate the latent distribution of user behavior, we assume that $z$ follow the Gaussian distribution with mean value $\sigma$ and standard deviation $\mu$. Then, we use the probabilistic decoder $p(x|z)$ to reconstruct the encoded representation into output data.

The difference between the input and output value of VAE is called the reconstruction loss, which consists of mean squared error (MSE) and KL-divergence. The reconstruction loss of VAE is computed as follows:

$$L(x) = -D_{KL}(q(z|x) \parallel p(z)) + E_{z\sim q(z|x)}[(log\, p(x|z))] \tag{1}$$

where, $q(z|x)$ is the encoder from the data layer to the hidden layer, $p(x|z)$ is the decoder from the hidden layer to the data layer. The loss function of VAE aims to reduce the KL divergence, making $q(z|x)$ closer to the prior distribution $p(z)$. The second term of Eq. 1 is the reconstruction error, making the reconstructed $p(x|z)$ close to the input distribution $p(x)$ as possible. By implementing the reconstruction process, VAE can learn the most important features from the original input data.

In the VAE pre-training process, we input the normal user data into VAE, and learn the important characteristics of normal users through training. Due to the learned distribution of user features, VAE can reconstruct normal user data. And when taking the abnormal behavioral data as input, the VAE cannot reconstruct the same data as the original input. That's because the VAE does not learn the features and distributions of abnormal users.

**BPNN Detection.** BPNN [23] is a supervised learning algorithm. We can view the BPNN algorithm as a nonlinear mapping from input to output. BPNN adjusts parameters by back propagation to obtain the closest result to the expected output. Figure 3 shows the topology of the BPNN.

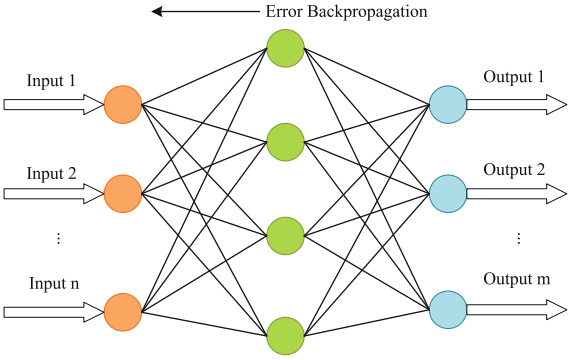



**Fig. 3.** Structure of BPNN.

The process for BPNN detection has three steps:

(1) Forward propagation: we use the representation from VAE as the input of BPNN. The input signal is propagated from the input layer, via the hidden layer, to the output layer. The output result is calculated by forward propagation. If the output of BPNN does not match the actual value, we go to the next step.
(2) Back propagation: we use the loss function for back propagation to adjust the weights between neurons in the different layers. By repeating step (2), the prediction output is as consistent as possible with the actual value.
(3) Detection: we can use the trained BPNN model to detect insider threats on the test dataset, and get the detection results finally.

**Algorithm 1.** VAE-BPNN Algorithm

**Input:** $X$-train dataset, $Y$-test dataset, $M$-dimension, $n$-number of epoch.
**Output:** detection result $S$.

Initialize the size of M and n.
$VAE \longleftarrow VAE(X, M)$
$VAE \longleftarrow VAE(Y, M)$
**for** i=1 to n **do**
    $BPNN \longleftarrow BPNN_i(X)$
**end for**
$S \longleftarrow BPNN_i(Y)$
**return** $result \longleftarrow S$

In the BPNN detection process, we train the BPNN model through the input user data, and then back-propagate the error, by continuously adjusting the weights between neurons to minimize the error value. Finally, we can use the trained BPNN model to detect whether the input user behavior is abnormal.

In a summary, we provide the pseudocode of the VAE-BPNN detection method, which is described in Algorithm 1. Where $X$ refers to the train dataset, $Y$ refers to the test dataset, $M$ refers to the data dimension before VAE pre-training, $n$ represents the training period, and $S$ represents the detection result on the test dataset.

## 4 Experiments

### 4.1 Experimental Dataset

In this study, we choose the CERT-IT dataset (r4.2 version) [24] as our experimental dataset. This dataset is a well-known synthetic insider threat dataset, its data is generated by Insider Threat Center at Carnegie Mellon University's Software Engineering Institute (SEI).

The CERT dataset consists of multiple files that provide logs of both background and malicious behavior of users in the organization. The file Logon.csv, Http.csv, Email.csv, File.csv, and Device.csv contain user activity information such as login/logout, website access, email, and internet browsing history, copy files to removable disk, etc.

Additionally, Psychometric.csv records scores about employee psychometric tests. And LDAP is a file that contains information about the organization's users and their roles. Because the experiment did not address the effects of factors such as employee psychometric scores and role information, psychometric.csv and LDAP were not used.

For different user behavior categories, we extract needed features from the dataset to create three feature sets. The eleven user behavior characteristics used in the experiment are shown in Table 1.

Furthermore, the dataset describes several scenarios where insider threats occur, as shown in Table 2. We conduct experiments for scenarios S1, S2 and S3 respectively. Besides the threat scenarios, the CERT dataset also provides a set of data for each threat scenario, including information such as the activity date and ID of each malicious person.

**Table 1.** User behavior characteristics used in the experiment.

| File | Characteristic | Specific user behavior |
|---|---|---|
| Logon.csv | logon | Number of logins |
| | logon out of work | Number of logins after work hours |
| | logon PC | Number of computers accessed |
| | logon PC out of work | Number of computers accessed after work hours |
| Device.csv | device | Number of device connections |
| | device out of work | Number of device connections after work hours |
| File.csv | file | Number of file operations |
| Email.csv | attachments | Number of email attachments |
| | email size | Average message size |
| | email to | Number of recipients |
| Http.csv | http | Number of visits to a website |

**Table 2.** CERT insider threat scenarios' description.

| Scenarios | Description |
|---|---|
| S1 | A user who has unusual behaviors uses a removable drive and uploads data to wikileaks.org, and leaves the company shortly thereafter |
| S2 | A user first looks for the company's competitors on the Internet, and then sells the company's secrets to competitors to gain benefits |
| S3 | The system administrator is disgruntled, and then retaliate against the company by sending out the confidential information by e-mail |

### 4.2 Experimental Environment

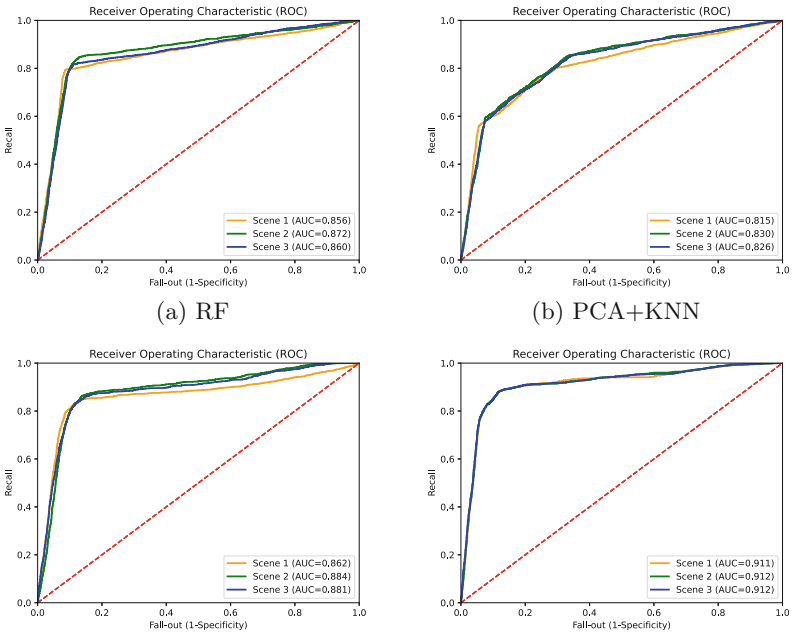
In the experiment, we use a single node host, which is equipped with an Intel I7-8700 CPU, 16 GB memory, and Ubuntu 18 operating system. Moreover, in terms of software, we use PyCharm Community 2017.3, Python 3.6, and Keras 2.2.4 to set up the software environment.

### 4.3 Experimental Results

In this part, the verification experiments are divided into two points: precision evaluation and effectiveness evaluation. And we choose three comparison methods, including random forest (RF), principal component analysis and k-nearest neighbor (PCA+KNN), auto-encoder and support vector machine (AE+SVM).

**Effectiveness Evaluation.** First, we conduct an experiment to evaluate the effectiveness of the VAE-BPNN method. And we detect insider threat in three described scenarios of the CERT dataset using the proposed method, RF method, PCA+KNN method, and AE+SVM method.

We choose the receiver operating characteristic (ROC) curve and the area under ROC curve (AUC) as the evaluation indicators. Figure 4 shows the ROC curves of the proposed method and other three comparison methods in Scenarios S1, S2, and S3.




(a) RF (b) PCA+KNN

(c) AE+SVM (d) VAE+BPNN

**Fig. 4.** The ROC curves of RF, PCA+KNN, AE+SVM, and VAE+BPNN.

In Fig. 4(d), the AUC values of the VAE-BPNN method are the highest, respectively 0.911, 0.912, and 0.912 in Scenarios S1, S2, and S3. The results show that the proposed method has higher precision while ensuring fewer false positive examples. That is because VAE learns the Gaussian distribution of normal user behavior, and fully retains the effective information of the data. Also, BPNN has better classification capability for detection anomalies. Also, we can observe that the ROC curves in (a) and (c) are very similar, and their AUC values is very close to the proposed method. They have similar detection performance.

With the same 1-specificity value (especially in the range of 0.1 to 0.4), the ROC curve in (b) has a lower recall rate than other three methods, indicating that it is less likely to detect anomalous users. That may be since PCA discards some important information during dimensionality reduction and the KNN algorithm is not suitable for detecting rare abnormal behavioral data.

For the three insider threat scenarios, we found little difference in the curves and AUC values of each method, which means that the different scenarios do not have much impact on the detection capability.

**Precision Evaluation.** Moreover, we conduct an experiment to evaluate the actual detection precision of our approach. And we choose Precision, Accuracy, Recall Rate, F1-Score as the evaluation metrics. The experimental results are shown in Fig. 5.

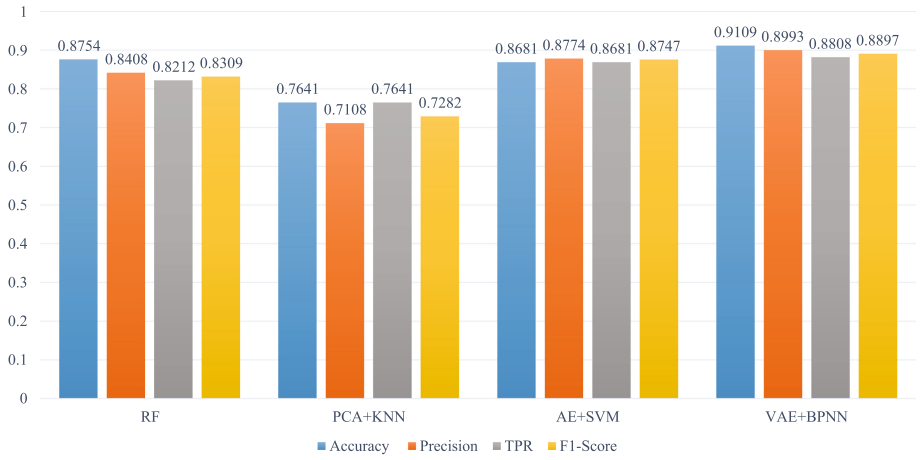



**Fig. 5.** Effectiveness comparison of different methods.

As shown in Fig. 5, the VAE-BPNN method is superior to the other three methods in Accuracy, Precision, TPR, and F1-Score. This is because VAE efficiently extracts the necessary user features and BPNN accurately detects anomalous behaviors. And, we can observe the performance of the AE+SVM method is very close to the proposed method. Since AE and VAE have similar structures, but VAE better learn the gap between normal and abnormal behavioral data.

RF algorithm has relatively good accuracy, but its precision and TPR are lower, as there are still some false negatives and false alarms. And the PCA+KNN method has the lowest detection accuracy and precision. That is because PCA can reduce the dimensionality of the input data, but miss some important features of malicious insiders. Therefore, the PCA+KNN detection method's performance is not good.

In summary, the overall detection performance of the VAE-BPNN approach is better than above comparison methods, and it is an effective method for insider threat detection.

## 5 Conclusion

In the paper, we proposed a novel insider threat detection method based on VAE and BPNN. To reduce the high false alarm rate, we used VAE to remove redundant user information, and then extract the compressed representation of user behavioral data. Through training VAE continuously, we learn the important characteristics of normal users, so that the encoded representation of abnormal behavior was significantly different from normal behavior. For the problem of poor generalization and detection performance, we used BPNN to fully learn user features before detecting, and finally discriminate whether user behavior was abnormal. According to the experimental results, the proposed VAE-BPNN detection method can achieve high accuracy and precision, which provides a solution for mitigating the insider threat issue.

For future work, we can fit the proposed method to real insider threat datasets to further verify its feasibility and performance. In addition, we can conduct more comparative experiments with a variety of emerging deep learning methods, and compare their performance with the proposed method within this paper.

**Acknowledgements.** This work was supported by the National Natural Science Foundation of China (No. 61962015) and the Guangxi Key Laboratory of Cryptography and Information Security Research Project (NO. GCIS202127), and the Innovation Project of GUET Graduate Education(No. 2022YCXS075).

## References

1. Verble, J.: The NSA and Edward Snowden: surveillance in the 21st century. ACM SIGCAS Comput. Soc. **44**(3), 14–20 (2014)
2. Kim, A., Oh, J., Ryu, J., Lee, J., Kwon, K., Lee, K.: SoK: a systematic review of insider threat detection. J. Wirel. Mob. Netw. Ubiquit. Comput. Dependable Appl. **10**(4), 46–67 (2019)
3. Homoliak, I., Toffalini, F., Guarnizo, J., Elovici, Y., Ochoa, M.: Insight into insiders and it: a survey of insider threat taxonomies, analysis, modeling, and countermeasures. ACM Comput. Surv. (CSUR) **52**(2), 1–40 (2019)
4. Cost of Insider Threats Report. https://www.ibm.com/security/digital-assets/services/cost-of-insider-threats/. Accessed 19 May 2022

5. Alsowail, R.A., Al-Shehari, T.: Empirical detection techniques of insider threat incidents. IEEE Access **8**, 78385–78402 (2020)
6. Liu, L., De Vel, O., Han, Q.L., Zhang, J., Xiang, Y.: Detecting and preventing cyber insider threats: a survey. IEEE Commun. Surv. Tutor. **20**(2), 1397–1417 (2018)
7. Yuan, S., Wu, X.: Deep learning for insider threat detection: review, challenges and opportunities. Comput. Secur. **104**, 102221 (2021)
8. Raveendran, R., Dhanya, K.A.: Covariance matrix method based technique for masquerade detection, pp. 1–5 (2014)
9. Taylor, P.J., et al.: Detecting insider threats through language change. Law Hum. Behav. **37**(4), 267 (2013)
10. Eberle, W., Graves, J., Holder, L.: Insider threat detection using a graph-based approach. J. Appl. Secur. Res. **6**(1), 32–81 (2010)
11. Seo, S., Kim, D.: Study on inside threats based on analytic hierarchy process. Symmetry **12**(8), 1255 (2020)
12. Qiu, W., Khong, A.W.H., Tay, W.P.: Hidden Markov model for masquerade detection based on sequence alignment. In: 2018 IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress, pp. 278–285. IEEE Computer Society (2018)
13. Ye, X., Han, M.M.: An improved feature extraction algorithm for insider threat using hidden Markov model on user behavior detection. Inf. Comput. Secur. (2020)
14. Wall, A., Agrafiotis, I.: A Bayesian approach to insider threat detection. J. Wirel. Mob. Netw. Ubiquit. Comput. Dependable Appl. **12**(2) (2021)
15. Chen, L., Aritsugi, M.: An SVM-based masquerade detection method with online update using co-occurrence matrix. In: Büschkes, R., Laskov, P. (eds.) DIMVA 2006. LNCS, vol. 4064, pp. 37–53. Springer, Heidelberg (2006). https://doi.org/10.1007/11790754_3
16. Le, D.C., Zincir-Heywood, A.N.: Machine learning based insider threat modelling and detection. In: IFIP/IEEE International Symposium on Integrated Network Management, IM, pp. 1–6. IFIP (2019)
17. Liu, L., De Vel, O., Chen, C., Zhang, J., Xiang, Y.: Anomaly-based insider threat detection using deep autoencoders. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 39–48. IEEE (2018)
18. Hu, T., Niu, W., Zhang, X., Liu, X., Lu, J., Liu, Y.: An insider threat detection approach based on mouse dynamics and deep learning. Secur. Commun. Netw. **2019** (2019)
19. Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., Robinson, S.: Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. In: Workshops at the Thirty-First AAAI Conference on Artificial Intelligence (2017)
20. Meng, F., Lou, F., Fu, Y., Tian, Z.: Deep learning based attribute classification insider threat detection for data security. In: Third IEEE International Conference on Data Science in Cyberspace, DSC, pp. 576–581. IEEE (2018)
21. Jiang, J., et al.: Anomaly detection with graph convolutional networks for insider threat and fraud detection. In: 2019 IEEE Military Communications Conference, MILCOM, pp. 109–114. IEEE (2019)
22. Doersch, C.: Tutorial on variational autoencoders. CoRR abs/1606.05908 (2016). http://arxiv.org/abs/1606.05908

23. Li, J., Cheng, J., Shi, J., Huang, F.: Brief introduction of back propagation (BP) neural network algorithm and its improvement. In: Jin, D., Lin, S. (eds.) Advances in Computer Science and Information Engineering, pp. 553–558. Springer, Cham (2012). https://doi.org/10.1007/978-3-642-30223-7_87
24. Glasser, J., Lindauer, B.: Bridging the gap: a pragmatic approach to generating insider threat data. In: 2013 IEEE Security and Privacy Workshops, pp. 98–104 (2013)

# VMT: Secure VANETs Message Transmission Scheme with Encryption and Blockchain

Shiyuan Xu[1,2], Xue Chen[1,3], Yunhua He[1(✉)], Yibo Cao[1,4], and Shang Gao[3]

[1] School of Information Engineering, North China University of Technology, Beijing, China
heyunhua@ncut.edu.cn

[2] Department of Computer Science, The University of Hong Kong, Pok Fu Lam, Hong Kong

[3] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong

[4] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China

**Abstract.** Vehicular ad-hoc networks (VANETs) is emerging as the most essential technology for securing traffic safety as well as enhancing traffic conditions. The confidentiality of messages transmitted in VANETs is of utmost priority, which has attracted extensive scholarly research. However, the security of the existing studied encryption schemes remains to be upgraded with a massy time overhead. Hence, we propose a lightweight encryption scheme for secure message transmission in VANETs with a private blockchain. Firstly, the encryption scheme we designed, called VMT, ensures the confidentiality and forward security of messages, while the system is sound and amnesic so that we are not entirely dependent on external servers. Meanwhile, the private blockchain with high privacy-preserving and flexible read/write access assures that merely permitted users to access the message. Furthermore, our scheme achieves excellent lightweight, thus suitable for VANETs with huge messages. Security analysis indicates that our scheme has firmly secured the transmission of messages in VANETs. Performance evaluation demonstrates that our scheme is lightweight and efficient, with encryption and decryption times of 7.00 ms and 7.67 ms, respectively.

**Keywords:** Message confidentiality · VANETs · Blockchain · Amnesia · Forward security · Security and privacy · Applied cryptography

S. Xu and X. Chen—have the same contribution in this paper with no particular order. Shiyuan Xu and Xue Chen completed this work when they were at North China University of Technology.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 244–257, 2022.
https://doi.org/10.1007/978-3-031-19208-1_21

## 1 Introduction

With the accelerated advancement in the Internet of Vehicles (IoV) [1–3] and Intelligent Transportation Systems (ITS) [4,5], VANETs [6,7] are deemed as the most potential technology for ensuring traffic safety and enhancing driving performance in the future [8], while provoking numerous studies [9,10,28–30]. In VANETs, Road Side Unit (RSU), which is deployed on the roadside or intersections according to its scope of telecommunication, communicates via IEEE 802.11 protocol [11] with the vehicle equipped wireless communication devices, i.e., On Board Unit (OBU) collaboratively at high speed and in real-time. The wireless communication of VANETs aims at supporting sophisticated driving assistance functions and tracking traffic situations timely to avoid traffic congestion [12] while achieving traffic safety. The prerequisite for VANETs to tackle driving sensations and traffic congestions is to assure a trustworthy and secure message dissemination service, whereas the confidentiality of messages are commonly neglected, leading to a considerable security vulnerability [13].

Security has gradually become the focus of scholarly research [25–27], several scholars noticed the security risks [23,24] of message propagation and conducted related researches in VANETs [14–19]. Cai [14] designed a ring sign-encryption scheme to reach user privacy protection and message confidentiality [20], but depends on third parties heavily. [15–17] employed Ciphertext Policy Attribute-Based Encryption (CP-ABE) for message encryption to assure message confidentiality and access control capability, however, their system overhead is burdensome. Due to the massive volume of messages in VANETs, we demand a lightweight system to maintain the system function properly during traffic congestion. To reach this, [18] proposed an arithmetically simple protocol to achieve lightweight yet lacking access control, leading to the illegal application of messages. [19] selected full homomorphic encryption to upgrade message security and performs data aggregation to fulfill the lightweight requirement while setting secondary decryption for access control, nonetheless, it has not considered forward security. So far, there is still no lightweight message transmission scheme that can guarantee forward security in VANETs.

To address the foregoing challenges, we propose a lightweight encryption scheme for secure message transmission in VANETs with a private blockchain. We devise VMT, including two servers that are not entirely trusted cooperate in encryption and decryption operation ensuring secure message transmission in VANETs. The VMT scheme maintains the confidentiality and integrity of messages and features strong soundness. Meanwhile, two servers forget immediately after completed the encryption and decryption actions, assuring the amnesia of the scheme. In addition, forward security is provided by the VMT scheme, which enables the previous messages with security even if the key is compromised. Further, due to the advantages of blockchain technology utilized in vehicular networks [21], private blockchain [22] is deployed in our scheme due to its superior security and rapid transaction. The OBU/RSU establishes its read/write privileges to ensure access control. Crucially, our scheme is lightweight to accommodate the massive messages transfer characteristics of VANETs.

Our main contributions are elaborated as follows:

1. We design a lightweight encryption scheme, called VMT, and employ it in VANETs, ensuring strong soundness, amnesia, messages confidentiality and forward security character.
2. Our scheme enables secure transmission of messages in VANETs without complete trust in the encryption server due to the innovative amnesia property as well as the VMT scheme.
3. Private blockchain is used to exert access control, with better transaction efficiency, lower-budget to deploy, more privacy-conscious with robust data security as well as with more flexible read/write permissions.
4. Security analysis and experimental evaluation indicate that our scheme is secure and lightweight. The encryption and decryption operations of one message take only 7.00 ms and 7.67 ms, less than existing systems.

The paper is organized as follows: The Sect. 2 is the threat models and design goals. We introduce our proposed scheme specified in Sect. 3. Security analysis is given in Sect. 4, and performance evaluation is plotted in Sect. 5. Then, we finish the literature review in Sect. 6. Finally, we conclude our paper in Sect. 7.

## 2 Threat Models and Design Goals

The acronyms utilized in our paper are shown in Table 1.

### 2.1 Threat Models

**Non-confidentiality of Messages.** The lack of confidentiality and integrity of messages transmitted in VANETs leads to privacy breaches.

**Completely Trust on Third Parties.** Fully trusting third parties can result in security deficiencies and the risk of manipulation like known-plaintext attack.

**Arbitrary Access.** Casually access to transport messages risks message abuse and flooding attacks.

**Time-consuming Encryption.** The overly burdensome encryption protects the confidentiality of messages whereas it causes the system to collapse during high traffic volumes.

**Table 1.** Glossary

| Acronym | Description | Acronym | Description |
|---|---|---|---|
| $(pk, sk)$ | Public-secret key pair | $ct_b$ | Ciphertext |
| $\vartheta$ | Non-interactive zero-knowledge proof | $l, l^*$ | Label |
| $\omega$ | Record | $F$ | Flag |
| $rn$ | Random number | $pw_b^*$ | Password |
| $Q, \mathbb{Q}$ | Prime order | $M, m_b$ | Message |

### 2.2 Design Goals

**Security of Message Transmission.** Our scheme demands to assure the confidentiality of OBU, OBU/RSU communication messages as well as security.

**Distrustful Reliance on External.** The messages are encrypted and transmitted without full trust in external servers.

**Access Control.** Only entities with permission are allowed to access the encrypted information.

**Lightweight Encryption.** The overhead of message encryption is required to be low enough to ensure proper operation during traffic congestion.

**Forward Security.** Prior encrypted messages are remaining security despite the loss of the key.

## 3 Our Proposed Encryption Scheme in VANETs

### 3.1 Overall

The communication model is shown in Fig. 1, consisting of five entities. **Road Side Units (RSUs)** are placed on roadsides or at crossroads based on communication coverage to monitor and broadcast current traffic conditions in real-time. **Vehicles/On-Board Units (OBUs)** on the vehicles convey road condition messages to others or RSUs. **Encryption-Decryption Server (EDS)** supplies services for encryption and decryption of messages in VANETs. **Cryptographic Auxiliary Server (CAS)** is the server assisting with EDS encryption and decryption. In this case, a superior level of security can be realized by encrypting messages through the joint collaboration of two servers (EDS, CAS). **Private Blockchain (PBC)** is owned by a single entity, which setting access and read/write permissions freely.

In our architecture, when OBU-1 collides with OBU-2, it will cause traffic jam around. In this case, the relevant OBU should promptly communicate the message to the RSU within the communication range in order to broadcast the message and notify the vehicles that are about to enter this roadway to take detour measures. However, in this process, the broadcast of this vehicle accident and blockage information may lead to privacy leakage risk since the adoption of a third-party server is also not entirely trustworthy. Arbitrary OBU access and the high-traffic nature of VANETs can lead to abuse risks and system breakdown. To address these challenges, we design a lightweight encryption scheme with forward security, access control through a private blockchain, and two servers to assist with encryption to reduce the risk of untrustworthy third parties.

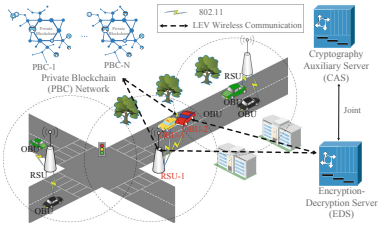



**Fig. 1.** System model.

Furthermore, we then denote message delivered between the RSU and the OBU assemble $M = \mathbb{Q}$, which $\mathbb{Q}$ is a group of prime order $q$ satisfies written multiplicatively and then set $\vartheta(Gen, Pok, Vf)$ as a Non-Interactive Zero-Knowledge (NIZK) proof. Further, we set two hash functions $H_{EDS}$ and $H_{CAS}$ as $H_{EDS}, H_{CAS} : \{0,1\}^* \to \mathbb{Q}$. Finally, we set the OBU/RSU password assemble $PW \leftarrow \{0,1\}^*$. We elaborate our scheme detailly as follows.

### 3.2 Setup and Key Extraction

Each OBU/RSU randomly acquires its password $pw_b^*$ and pseudo-ID $U_{ID}^E$, where $E$ represents its entity type as OBU/RSU, which is used for subsequent encryption and decryption. Simultaneously, the blockchain configures initially for no read/write access by anyone except the owner. The two mutually collaborating servers used for encryption extract their key pairs at this step as well.

*VMT.Setup and KeyExt:* After input the security parameter $\lambda$, the algorithm sets the public parameter $p$. Then, through calling $\vartheta.Gen$, it gets the parameter $a$ and also obtains $Q \leftarrow \mathbb{Q}$. After that, according to Algorithm 1, it can easily get the $(pk_{EDS}, sk_{EDS})$ and $(pk_{CAS}, sk_{CAS})$.

**Algorithm 1.** *VMT. Setup and KeyExt*

**Input:** Security parameter $\lambda$
**Output:** $(a, \vartheta)$, $(pk_{EDS}, sk_{EDS})$ and $(pk_{CAS}, sk_{CAS})$

1: Procedure *VMT. Setup and KeyExt*
2: Set public parameter $p \leftarrow Setup(1^\lambda)$
3: Set $a \leftarrow \vartheta.Gen(1^\lambda)$ and $Q \leftarrow \mathbb{Q}$
4: **Return** $(a, \vartheta)$
5: Set $pk_{EDS} \leftarrow \perp$ and $sk_{EDS} \leftarrow \mathbb{Z}_q$ ▷ where $\perp$ means empty
6: **Return** $(pk_{EDS}, sk_{EDS})$
7: Set $x \leftarrow \mathbb{Z}_q$ and $pk_{CAS} \leftarrow Q^x$
8: Set $sk_{CAS} \leftarrow x$
9: **Return** $(pk_{CAS}, sk_{CAS})$

### 3.3 Encryption

As shown in Fig. 1, $OBU_1$ and $OBU_2$ collide, causing traffic congestion on main road. $OBU_1$ and/or $OBU_2$ reports the incident by communicating with the nearest $RSU_1$ within its telecommunication coverage, with aiming that $RSU_1$ promptly broadcasts to the vehicle behind reminding it to re-route as soon as possible. For convenience, in the following description we refer to the reported message as $M$, the entity sending the message as $OBU_S$, and that receiving the message as $RSU_R$. Notably, our scheme is suitable for communication between arbitrary two sides in VANETs.

*Step* 1. $OBU_S$ generates $M$ which is sent to EDS with pseudo-ID $U_{ID}^E$ and $pw_b^*$ to request encryption service.

*Step* 2. EDS and CAS jointly encrypt the message $M$ to produce the $\omega$.

**Algorithm 2.** $VMT.\ EDS-Encryption((l^*,\omega),\perp)$

**Input:** EDS secret key $sk_{EDS}$, CAS secret key $sk_{CAS}$, password $pw_b^*$, and message $M$
**Output:** $((l^*,\omega),\perp)$

Procedure $VMT.\ EDS-Encryption$
  Set label $l=(l_{EDS},l_{CAS})$
  Generate EDS and CAS random number $rn_{EDS},rn_{CAS}$
  **if** $l=\perp$ **then** ▷ where $\perp$ means empty
    $rn_{EDS}\leftarrow\{0,1\}^\lambda$
  **else**
    Denote $l$ as $(rn_{EDS},rn_{CAS})$
  **end if**
  Set $H_{EDS,m_0}\leftarrow H_{EDS}(pw_0^*,rn_{EDS},0)$
  Set $H_{EDS,m_1}\leftarrow H_{EDS}(pw_1^*,rn_{EDS},1)$
  Set Ciphertext $Ct_b=(H_1(U_{ID}^E),H_2(pw_b^*))\cdot m_b$
  Compute $\omega=(Ct_0H_{EDS,m_0},Ct_1H_{EDS,m_1}M)$
  Compute label $l^*=(rn_{EDS},rn_{CAS})$
  **Return** $((l^*,\omega),\perp)$

$VMT.EDS-Encryption((l^*,\omega),\perp)$*:* The EDS inputs secret key $sk_{EDS},sk_{CAS}$, password $pw_b^*$ and message $M$. This protocol is as follows. We initially set the label $l=(l_{EDS},l_{CAS})$ and then generate random numbers $rn_{EDS}$ and $rn_{CAS}$ of EDS and CAS, respectively. Further, if $l=\perp$, it will $rn_{EDS}\leftarrow\{0,1\}^\lambda$, others will denote $l$ as $(rn_{EDS},rn_{CAS})$. After that, we set $H_{CAS,m_0}$ and $H_{CAS,m_1}$ through this algorithm, and the two parties create the ciphertext $Ct_b=(H_1(U_{ID}^E),H_2(pw_b^*))\cdot m_b$, where $b\in\{0,1\}$. Finally, we compute $\omega=(Ct_0H_{EDS,m_0},Ct_1H_{EDS,m_1}M)$, the label $l^*=(rn_{EDS},rn_{CAS})$ and output them, as Algorithm 2.

$VMT.\ CAS-Encryption(\perp,(rn_{CAS},Ct,\mu))$*:* The input is same as Algorithm 2 and we also set the label $l=(l_{EDS},l_{CAS})$, random numbers $rn_{EDS}$ and $rn_{CAS}$. Moreover, if $l=\perp$, it will $rn_{EDS}\leftarrow\{0,1\}^\lambda$, else name $l$ as $(rn_{EDS},rn_{CAS})$. Then, we set $H_{CAS,m_0}$ and $H_{CAS,m_1}$ according to this algorithm and set ciphertext $Ct=(Ct_0,Ct_1)\leftarrow(H^x_{CAS,m_0},H^x_{CAS,m_1})$. Then, if $\exists x\ s.t.\ (Ct_0,Ct_1,pk_{CAS})=(H^x_{CAS,m_0},H^x_{CAS,m_1},Q^x)$, we denote $\delta=x$. Finally, we process $\mu\leftarrow\vartheta.PoK(a,\delta)$ with outputting $(\perp,(rn_{CAS},Ct,\mu))$. Details are shown in Algorithm 3.

*Step* 3. The EDS transmits the encrypted message $\omega$ to $OBU_S$ and deletes $M$.

*Step* 4. $OBU_S$ uploads the password $pw_b^*$ to the private blockchain as well as grants read access to $RSU_R$.
*Step* 5. $OBU_S$ transmits $\omega$ to $RSU_R$ via IEEE 802.11 protocol.

### 3.4 Decryption

After $RSU_R$ receives the encrypted message $\omega$ transmitted by $OBU_S$, it will decrypt $\omega$ to obtain the message $M$. The detailed decryption procedure is shown below.

**Algorithm 3.** $VMT.\ CAS-Encryption(\perp, (rn_{CAS}, Ct, \mu))$

**Input:** EDS secret key $sk_{EDS}$, CAS secret key $sk_{CAS}$, password $pw_b^*$, and message $M$
**Output:** $(\perp, (rn_{CAS}, Ct, \mu))$

1: Procedure $VMT.\ CAS-Encryption$
2: Set label $l = (l_{EDS}, l_{CAS})$
3: Generate EDS and CAS random number $rn_{EDS}, rn_{CAS}$
4: **if** $l = \perp$ **then** ▷ where $\perp$ means empty
5: $rn_{CAS} \leftarrow \{0,1\}^{\lambda}$
6: **else**
7: Denote $l$ as $(rn_{EDS}, rn_{CAS})$
8: **end if**
9: Set $H_{CAS,m_0} \leftarrow H_{CAS}(rn_{CAS}, 0)$
10: Set $H_{CAS,m_1} \leftarrow H_{CAS}(rn_{CAS}, 1)$
11: Set Ciphertext $Ct = (Ct_0, Ct_1) \leftarrow (H^x_{CAS,m_0}, H^x_{CAS,m_1})$
12: **if** exist $x$, satisfies $(Ct_0, Ct_1, pk_{CAS}) = (H^x_{CAS,m_0}, H^x_{CAS,m_1}, Q^x)$
13: Set $\delta = x$
14: **end if**
15: Set $\mu \leftarrow \vartheta.PoK(a, \delta)$
16: **Return** $(\perp, (rn_{CAS}, Ct, \mu))$

*Step* 1. $RSU_R$ accesses $OBU_S$'s private blockchain to obtain the password $pw_b^*$ for decryption.
*Step* 2. $RSU_R$ delivers the password $pw_b^*$ and encrypts message $\omega$ to EDS to request decryption service.
*Step* 3. EDS and CAS validate the password $pw_b^*$, then jointly decrypt it if it is correct, with the following algorithm.

$VMT.\ EDS-Decryption((sk_{EDS}, pw_b^*, \omega), \perp)$*:* After a user logs in with his/her password $pw_b^*$, the EDS will search its record $\omega$ and label $l$ as input and also EDS inputs its secret key $sk_{EDS}$. Firstly, we recall the record $\omega = (Ct_0 H_{EDS}, m_0, Ct_1 H_{EDS,m_1} M)$ and label $l = (rn_{EDS}, rn_{CAS})$. Moreover, we set $H_{CAS,m_0}$, $H_{CAS,m_1}$, $H_{EDS,m_0}$, $H_{EDS,m_1}$, respectively. After that, it will denote two ciphertexts $Ct_0$ and $Ct_1$. Further, if the flag $F = TRUE$ and if $\exists x$ s.t. $(Ct_0, Ct_1, pk_{CAS}) = (H^x_{CAS,m_0}, H^x_{CAS,m_1}, Q^x)$, we set $\delta = x$ and assign $M \leftarrow (\omega_1 Ct_1 H_{EDS,m_1})$. Else, it will let $M = \perp$ and finally with $((sk_{EDS}, pw_b^*, \omega), \perp)$ as return. We elaborate them in Algorithm 4.

**Algorithm 4.** *VMT. EDS − Decryption*$((sk_{EDS}, pw_b^*, \omega), \perp)$

**Input:** EDS secret key $sk_{EDS}$, label $l$, record $\omega$, password $pw_b^*$, and flag $F$
**Output:** $((sk_{EDS}, pw_b^*, \omega), \perp)$

Procedure *VMT. EDS − Decryption*
  Recall $\omega = (Ct_0 H_{EDS,m_0}, Ct_1 H_{EDS,m_1} M)$
  Recall $l = (rn_{EDS}, rn_{CAS})$
  Set $H_{CAS,m_0} \leftarrow H_{CAS}(rn_{CAS}, 0)$
  Set $H_{CAS,m_1} \leftarrow H_{CAS}(rn_{CAS}, 1)$
  Set $H_{EDS,m_0} \leftarrow H_{EDS}(pw_0^*, rn_{EDS}, 0)$
  Set $H_{EDS,m_1} \leftarrow H_{EDS}(pw_1^*, rn_{EDS}, 1)$
  Set Ciphertext $Ct_0 \leftarrow \omega_0 H_{EDS,m_0}$ and $Ct_1 \leftarrow \omega_1 H_{EDS,m_1}$
  **if** $F = TRUE$ **then**
    **if** exist $x$ satisfies **then** $(Ct_0, Ct_1, pk_{CAS}) = (H^x_{CAS,m_0}, H^x_{CAS,m_1}, Q^x)$
      Set $\delta = x$ and $M \leftarrow (\omega_1 Ct_1 H_{EDS,m_1})$
    **end if**
  **else**
    $M = \perp$ ▷ where $\perp$ means empty
  **end if**
  **Return** $((sk_{EDS}, pw_b^*, \omega), \perp)$

**Algorithm 5.** *VMT. CAS − Decryption*$(\perp, (sk_{CAS}))$

**Input:** label $l$, record $\omega$, and flag $F$
**Output:** $(\perp, (sk_{CAS}))$

Procedure *VMT. CAS − Decryption*
  Recall Ciphertext $Ct_0 \leftarrow \omega_0$
  Recall $l = (rn_{EDS}, rn_{CAS})$
  Set $H_{CAS,m_0} \leftarrow H_{CAS}(rn_{CAS}, 0)$
  Set $H_{CAS,m_1} \leftarrow H_{CAS}(rn_{CAS}, 1)$
  **if** $Ct_0 = H^x_{CAS,m_0}$ **then**
    $F = TRUE$ and $Ct_1 = H^x_{CAS,m_1}$
    **if** exist $x$ satisfies **then** $(Ct_0, Ct_1, pk_{CAS}) = (H^x_{CAS,m_0}, H^x_{CAS,m_1}, Q^x)$
      Set $\delta = x$ and obtain $sk_{CAS}$
    **end if**
  **else**
    Set $F = FALSE$ and $M = \perp$ ▷ where $\perp$ means empty
  **end if**
  **Return** $(\perp, (sk_{CAS}))$

*VMT. CAS − Decryption*$(\perp, (sk_{CAS}))$*:* The input in this step is label $l$, record $\omega$ and flag $F$. To begin, it recalls ciphertext $Ct_0 \leftarrow \omega$ and $l = (rn_{EDS}, rn_{CAS})$. After that, we denote $H_{CAS,m_0}$ and $H_{CAS,m_1}$. Then, if $\exists x$ *s.t.* $(Ct_0, Ct_1, pk_{CAS}) = (H^x_{CAS,m_0}, H^x_{CAS,m_1}, Q^x)$, we set $\delta = x$ and obtain $sk_{CAS}$. Else, it will set $F = TRUE$ and $M = \perp$. Lastly, this algorithm will return $(\perp, (sk_{CAS}))$ and details expounded in Algorithm 5.

*Step* 4. EDS delivers the decrypted message $M$ to $RSU_R$, and then deletes it.
*Step* 5. $OBU_S$ closes the read access of $RSU_R$ to its private blockchain.

## 4 Security Analysis

### 4.1 Correctness of VMT

Our VMT scheme is correct since each authentically generated record is decrypted successfully with the correct password input. The correctness of VMT also concludes soundness and forward security, which will show later. We then have the $Corr^{VMT}(1^\lambda)$ as: $Prob[Corr^{VMT}(1^\lambda) = 1] \leq |Prob[Soun_{S^\lambda}^{VMT}(1^\lambda) = 1] + |Prob[Forw_{S^\lambda}^{VMT,m_0}(1^\lambda) = 1] - Prob[Forw_{S^\lambda}^{VMT,m_1}(1^\lambda) = 1]||$.

### 4.2 Soundness of VMT

The proposed VMT scheme satisfies soundness, ensuring that after an honest EDS and a CAS produced a record and encrypted message, they can be regained through using correct passwords in the decryption protocol. In addition, soundness also satisfies that if adversaries execute the decryption protocol by incorrect passwords, the VMT scheme will output the flag $F = ERROR$.

**Definition 1 (Soundness).** VMT scheme satisfies soundness whenever each Probabilistic Polynomial Times(PPT) attacker $S^\lambda = (S_1^\lambda, S_2^\lambda, S_3^\lambda)$, it has one negligible function $Negl_{Soun}(\lambda)$ where $Prob[Soun_{S^\lambda}^{VMT}(1^\lambda) = 1] \leq Negl_{Soun}(\lambda)$.

### 4.3 Message Confidentiality

Message confidentiality requires that PPT attacker $S^\lambda$ has no ability to figure out whether one record $\omega^*$ has encrypted message $m_0^*$ or $m_1^*$, even if they were chosen by $S^\lambda$.

(1) Query phase: We formalize message confidentiality $Conf_{S^\lambda}^{VMT,m_b}(1^\lambda)$ including two attackers $S^\lambda = (S_1^\lambda, S_2^\lambda)$ as the EDS and one honest challenger to be CAS. Then attacker $S_1^\lambda$ will access the VMT encryption, decryption, and key update oracles respectively. Finally, $S_1^\lambda$ will output the secret key of EDS $sk_{EDS}^*$, one password distribution $\xi$ as well as messages $m_0$, $m_1$ and a state $st$.
(2) Challenge phase: Challenger $C$ selects one password $pw^*$ randomly and then writes $m_b^*$ in the record $\omega^*$ by using $sk_{EDS}^*$ and $pw^*$. In this phase, there is no communication transcript given to $S^\lambda$.
(3) Guess phase: $S_2^\lambda$ owns $\omega^*$ and $S_2^\lambda$ has to guess whether $m_0^*$ or $m_1^*$ has been encrypted by one guess $b^{'}$.

**Definition 2 (Message Confidentiality).** Our VMT scheme satisfies message confidentiality, if all PPT adversaries $S^\lambda = (S_1^\lambda, S_2^\lambda)$, it has one negligible function $Negl_{Conf}(\lambda)$ where $|Prob[Conf_{S^\lambda}^{VMT,m_0}(1^\lambda) = 1] - Prob[Conf_{S^\lambda}^{VMT,m_1}(1^\lambda) = 1]| \leq 2\sum_{i=1}^{N} p_i + Negl_{Conf}(\lambda)$.

### 4.4 Amnesia of VMT

Amnesia attribute ensures hiding passwords and encrypting messages against baleful CAS, which does not guarantee the anonymity of the incomplete end user. Thus, we formalize the amnesia of VMT as $Amne_{S^\lambda}^{VMT,m_b}(S^\lambda)$, which consists of three PPT attackers $S^\lambda = (S_1^\lambda, S_2^\lambda, S_3^\lambda)$ as hostile CAS and one challenger $C$ to be an honest EDS.

(1) Query phase: Firstly, $S_1^\lambda$ interacts with the challenger $C$ of the protocols for encryption and decryption. Then, the CAS outputs the protocols given to $S^\lambda$. $S_1^\lambda$ will output $(pw_0^*,\ m_0^*,\ pw_1^*,\ m_1^*)$ together with giving the state $st$ to $S_2^\lambda$.

(2) Challenge phase: With owning the challenge password $pw_b^*$ and message $m_b^*$, the challenger $C$, being one CAS, will use the encryption protocol through $pw_b^*$ and $m_b^*$ together with the attacker $S_2^\lambda$ as a CAS. Finally, $C$ outputs and sends the record $T^*$ to $S_3^\lambda$, then $S_2^\lambda$ outputs and sends a state $st$ to $S_3^\lambda$.
(3) Guess phase: $S_3^\lambda$ still has the ability to interact with $C$ but the decryption oracle will not give the output to $S^\lambda$ with inputting $pw_0^*$ and $pw_1^*$. Thus, $S^\lambda$ cannot win the challenge and finally, having obtained the chosen $(pw_b^*, m_b^*)$ in the challenge phase, adversary $S_3^\lambda$ outputs the guess $b^{'}$.

**Definition 3 (Amnesia).** If all PPT attackers $S^\lambda = (S_1^\lambda, S_2^\lambda, S_3^\lambda)$ existing, scheme satisfies the negligible function $Negl_{Amne}(\lambda)$ where $|Prob[Amne_{S^\lambda}^{VMT,m_0}(1^\lambda) = 1] - Prob[Amne_{S^\lambda}^{VMT,m_1}(1^\lambda) = 1]| \leq Negl_{Amne}(\lambda)$.

### 4.5 Access Control

Due to the feature of heavy flows in VANETs, we demand access control for messages between OBUs. Meanwhile, access control prevents the risk of malicious user abused the messages, which are encrypted through VMT for secure transmission, allowing the only user with the correct password to access the messages by decrypting them via EDS. The private blockchain is owned by the entity that encrypts the message, while the transactions are incredibly speedy with solid trust. We also utilize the private blockchain to store the correct passwords with timely updates, while read/write access is only available to permit entities to enforce message access control.

### 4.6 Forward Security

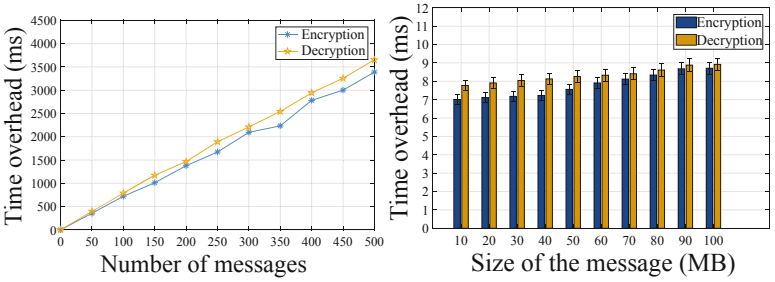
**Definition 4 (Forward Security).** We have to avoid the leakage of messages in VANETs scenarios even if the attackers have mastered the secret keys. Our proposed scheme is forward secure whenever any two PPT adversaries $S^\lambda = (S_1^\lambda, S_2^\lambda)$, they satisfy one negligible function $Negl_{Forw}(\lambda)$ where $|Prob[Forw_{S^\lambda}^{VMT,m_0}(1^\lambda) = 1] - Prob[Forw_{S^\lambda}^{VMT,m_1}(1^\lambda) = 1]| \leq Negl_{Forw}(\lambda)$.

## 5 Performance Evaluation

To evaluate the peculiarity of lightweight of our VMT scheme of VANETs, we tested the time overhead through digital simulation experiments utilizing data that is extremely proximate to practical applications. Meanwhile, we compared our VMT scheme with existing schemes [15,19] in order to highlight our superiority. The experiments based on the C++ language and simulated on Windows 10 with Intel(R) Core(TM) i7-10875H CPU @ 2.60 GHz, 16 GB RAM. The experimental results on behalf of 40 trials on average are demonstrated as follows.

Figure 2 depicts the impact of the number and message size in VMT on the time overhead of encryption and decryption operations. As shown in Fig. 2(a),

the time overhead goes up somewhat marginally and maintains relatively steady along with the growing size of message. Figure 2(b) demonstrates the effect of message size. We notice that time for encryption and decryption operations rise slightly as the message size expands, with the overall time maintaining stability.



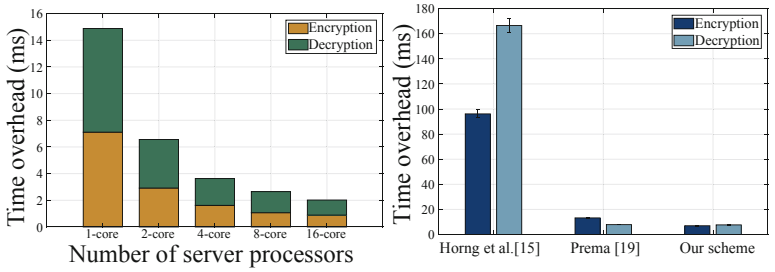

(a) Impact of number. (b) Impact of size.

**Fig. 2.** Compare the impact of message number and size.

Figure 3 illustrates the time overhead of VMT with a different number of processor cores and compares it with [15,19]. From Fig. 3(a), the time overhead for encryption and decryption decreases significantly when the number of server

**Table 2.** Security comparison with current schemes in properties.

| Schemes | Soundness | Message confidential | Amnesia | Access control | Forward security | Lightweight | Security mechanism |
|---|---|---|---|---|---|---|---|
| Horng [15] | × | √ | × | √ | √ | × | CP-ABE |
| Xia [16] | × | √ | × | √ | × | × | CP-ABE |
| Liu [17] | × | √ | × | √ | × | × | CP-ABE, PRE |
| Li [18] | × | √ | × | × | × | √ | Authentication |
| Our scheme | √ | √ | √ | √ | √ | √ | VMT |




(a) Comparison of processors number. (b) Comparison with other schemes.

**Fig. 3.** Comparison of processors number and comparison with other schemes.

processors increases. Higher core processors can be deployed at frequent traffic congestions to enhance processing speed. In Fig. 3(b), we compare the time overhead with existing systems using a 1-core processor as a premise. It is apparent that our scheme fulfills the lightweight and outperforms existing schemes [15,19] markedly.

## 6 Related Work

VANETs is emerging as a critical technology for enhancing traffic conditions and driving safety, which requires frequent communication between OBU and RSU. The issue of communication security in VANETs networks has attracted widespread attention as well as catalyze numerous studies. As shown in Table 2, we conduct a security comparison with existing schemes [15–18] with an analysis of their strengths and weaknesses, respectively. To solve the challenge of data confidentiality in VANETs, [15] introduced an identity-based scheme on revocable CP-ABE, where encryption and decryption operations were outsourced to the cloud for reducing the burden on OBU. As well, the scheme guaranteed user privacy and access control, however it performed poorly in terms of amnesia and system lightweight. The protocol proposed in [18] applied hash functions and exclusive-OR operations with system lightweight capabilities, however, it did not promise forward security and access control of messages. [16] employed CP-ABE for protecting message confidentiality while [17] combined CP-ABE with Proxy Re-Encryption (PRE) for an enhancement in time overhead. Whereas, none of them ensure the forward security of the messages with the lightweight of the system. Moreover, none of the existing schemes exhibit strong soundness and amnesia. Massive amounts of messages are propagated in VANETs networks, especially during traffic congestion, hence lightweight is essential. Our designed lightweight VMT scheme provides strong soundness, amnesia, access control, and forward security properties while ensuring message confidentiality, superior to existing schemes.

## 7 Conclusion

In this paper, we propose a lightweight encryption scheme, VMT, for securing the transmission of high-volume messages in VANETs. Our scheme guarantees the correctness and strong soundness for the reliability of encryption services, while VMT amnesia ensures that the encryption is not dependent on third-party servers. Simultaneously, VMT enables message confidentiality and forward security, securing transmission of messages and previous messages in VANETs even if keys are compromised. In addition, multiple private blockchains owned by OBU/RSU entities adopt to fulfill message access control. Crucially, our scheme is lightweight, dramatically contributing to the system's stability under traffic congestion. Security analysis and comprehensive experimental evaluation show that VMT secures message transmission in VANETs with lower-budget overhead, excels to existing systems. Nevertheless, our scheme is unable to resist quantum computing attacks, which will be an urgent priority for future work.

**Acknowledgement.** This work was supported in part by the R&D Program of Beijing Municipal Education Commission under Grant KM202010009010, in part by the Yunnan Key Laboratory of Blockchain Application Technology under Grant 2021105AG070005 (YNB202102), in part by the Beijing Municipal Natural Science Foundation under Grant M21029 and in part by the National Key Research and Development Program of China under Grant 2018YFB1800302.

## References

1. Qian, Y., Zhang, Y., Fortino, G., Miao, Y., Hu, L., Hwang, K.: Security-enhanced content caching for the 5G-based cognitive internet of vehicles. IEEE Netw. **35**(2), 40–45 (2021)
2. Wang, X., Ning, Z., Hu, X., Wang, L., Guo, L., Hu, B., et al.: Future communications and energy management in the internet of vehicles: toward intelligent energy-harvesting. IEEE Wirel. Commun. **26**(6), 87–93 (2019)
3. Xu, S., Chen, X., Wang, C., He, Y., Xiao, K., Cao, Y.: A lattice-based ring signature scheme to secure automated valet parking. In: Liu, Z., Wu, F., Das, S.K. (eds.) WASA 2021. LNCS, vol. 12938, pp. 70–83. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86130-8_6
4. Qi, X., Mei, G., Piccialli, F.: Resilience evaluation of urban bus-subway traffic networks for potential applications in IoT-based smart transportation. IEEE Sens. J. **21**, 25061–25074 (2020)
5. Mollah, M.B., et al.: Blockchain for the internet of vehicles towards intelligent transportation systems: a survey. IEEE Internet Things J. **8**(6), 4157–4185 (2021)
6. Zhang, J., Zhang, Q.: On the security of a lightweight conditional privacy-preserving authentication in VANETs. IEEE Trans. Inf. Forensics Secur. (2021). Early access
7. Liu, Z., Liu, Z., Zhang, L., Lin, X.: MARP: a distributed MAC layer attack resistant pseudonym scheme for VANET. IEEE Trans. Dependable Secur. Comput. **17**(4), 869–882 (2020)
8. Khelifi, H., et al.: Named data networking in vehicular ad hoc networks: state-of-the-art and challenges. IEEE Commun. Surv. Tutorials **22**(1), 320–351 (2020)
9. Modesto, F., Boukerche, A.: Towards integrating public transit bus systems into urban and intelligent vehicular networks. In: WCNC 2019, pp. 1–6 (2019)
10. Jalooli, A., Song, M., Xu, X.: Delay efficient disconnected rsu placement algorithm for VANET safety applications. In: WCNC 2017, pp. 1–6 (2017)
11. Lanante, L., Roy, S.: Analysis and optimization of channel bonding in dense IEEE 802.11 WLANs. IEEE Trans. Wirel. Commun. **20**(3), 2150–2160 (2021)
12. Tao, Y., Sun, P., Boukerche, A.: A novel travel-delay aware short-term vehicular traffic flow prediction scheme for VANET. In: WCNC 2019, pp. 1–6 (2019)
13. Di Pietro, R., Guarino, S., Verde, N.V., Domingo-Ferrer, J.: Security in wireless ad-hoc networks-a survey. Comput. Commun. **51**, 1–20 (2014)
14. Cai, Y., Zhang, H., Fang, Y.: A conditional privacy protection scheme based on ring signcryption for vehicular ad hoc networks. IEEE Internet Things J. **8**(1), 647–656 (2021)
15. Horng, S.-J., Lu, C.-C., Zhou, W.: An identity-based and revocable data-sharing scheme in VANETs. IEEE Trans. Veh. Technol. **69**(12), 15933–15946 (2020)
16. Xia, Y., Chen, W., Liu, X., Zhang, L., Li, X., Xiang, Y.: Adaptive multimedia data forwarding for privacy preservation in vehicular ad-hoc networks. IEEE Trans. Intell. Transp. Syst. **18**(10), 2629–2641 (2017)

17. Liu, X., Chen, W., Xia, Y.: Security-aware information dissemination with fine-grained access control in cooperative multi-RSU of VANETs. IEEE Trans. Intell. Transp. Syst. (2020). Early access
18. Li, X., Liu, T., Obaidat, M.S., Wu, F., Vijayakumar, P., Kumar, N.: A lightweight privacy-preserving authentication protocol for VANETs. IEEE Syst. J. **14**(3), 3547–3557 (2020)
19. Prema, N.K.: Efficient secure aggregation in VANETs using fully homomorphic encryption (FHE). Mob. Netw. Appl. **24**(2), 434–442 (2019)
20. Yuen, T.H., Esgin, M.F., Liu, J.K., Au, M.H., Ding, Z.: *DualRing*: generic construction of ring signatures with efficient instantiations. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12825, pp. 251–281. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84242-0_10
21. Xu, S., Chen, X., He, Y.: EVchain: an anonymous blockchain-based system for charging-connected electric vehicles. Tsinghua Sci. Technol. **26**(6), 845–856 (2021)
22. Jo, M., Hu, K., Yu, R., Sun, L., Conti, M., Du, Q.: Private blockchain in industrial IoT. IEEE Netw. **34**(5), 76–77 (2020)
23. Chen, X., Xu, S., Qin, T., Cui, Y., Gao, S., Kong, W.: AQ-ABS: anti-quantum attribute-based signature for EMRs sharing with blockchain. In: WCNC 2022. pp. 1176–1181 (2022)
24. Chen, X., Xu, S., He, Y., Cui, Y., He, J., Gao, S.: LFS-AS: lightweight forward secure aggregate signature for e-health scenarios. In: ICC 2022. pp. 1239–1244 (2022). Early access
25. Zhu, S., Li, W., Li, H., Tian, L., Luo, G., Cai, Z.: Coin hopping attack in blockchain-based IoT. IEEE Internet Things J. **6**(3), 4614–4626 (2019)
26. Zhu, S., Cai, Z., Hu, H., Li, Y., Li, W.: zkCrowd: a hybrid blockchain-based crowdsourcing platform. IEEE Trans. Industr. Inf. **16**(6), 4196–4205 (2020)
27. Cai, Z., Zheng, X.: A private and efficient mechanism for data uploading in smart cyber-physical systems. IEEE Trans. Netw. Sci. Eng. **7**(2), 766–775 (2020)
28. Xu, H., Cai, Z., Li, R., Li, W.: Efficient CityCam-to-edge cooperative learning for vehicle counting in ITS. IEEE Trans. Intell. Transp. Syst. (2022)
29. Xiong, Z., Cai, Z., Han, Q., Alrawais, A., Li, W.: ADGAN: protect your location privacy in camera data of auto-driving vehicles. IEEE Trans. Industr. Inf. **17**(9), 6200–6210 (2021)
30. Cao, Y., Xu, S., Chen, X., He, Y., Jiang, S.: A forward-secure and efficient authentication protocol through lattice-based group signature in VANETs scenarios. Comput. Netw. **124**, 109–149 (2022)

# Robust Truth Discovery Against Multi-round Data Poisoning Attacks

Hongniu Zhang[1], Mohan Li[1(✉)], Yanbin Sun[1], and Guanqun Qu[2]

[1] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, Guangdong, China
2112006266@e.gzhu.edu.cn, {limohan,sunyanbin}@gzhu.edu.cn
[2] GDCERT/CC, Guangzhou, Guangdong, China

**Abstract.** Crowdsensing systems collect multidimensional and heterogeneous data using sensing devices of individuals (called workers). However, there are often conflicts between data from multiple sources. Truth discovery try to resolve the conflicts by evaluating the trustworthiness of each source to find the ground truth. However, recent works have shown that truth discovery is vulnerable to data poisoning attacks. By submitting carefully crafted malicious data, attackers can launch multi-round iterations of hiding and attacking to deceive trustworthiness from truth discovery. This can result in long-term, insidious, and damaging data poisoning attacks. To this end, it is necessary to study how to improve the robustness of truth discovery. In this paper, we study high robust truth discovery methods against multi-round data poisoning attacks in crowdsensing systems, and attempt to detect potentially malicious workers. First, we implement one of the most widely used truth discovery frameworks, and verify its vulnerability to multi-round data poisoning attacks. Second, we propose HRTDMD, a high robust truth discovery method with malicious worker detection. Finally, we conduct experiments on real dataset and verify the effectiveness of the proposed method.

**Keywords:** Truth discovery · Crowdsensing systems · Data poisoning

This work is funded by the Guangdong Basic and Applied Basic Research Foundation (No. 2021A1515012307, 2020A1515010450), the National Key Research and Development Plan (Grant No. 2020YFB2009503), Guangzhou Basic and Applied Basic Research Foundation (No. 202102021207, 202102020867), the National Natural Science Foundation of China (No. 62072130, 61702223, 61702220), the Guangdong Province Key Area R&D Program of China (No. 2019B010137004), Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019), and Guangdong Higher Education Innovation Group (No. 2020KCXTD007) and Guangzhou Higher Education Innovation Group (No. 202032854).


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 258–270, 2022.
https://doi.org/10.1007/978-3-031-19208-1_22

## 1 Introduction

Crowdsourcing is widely favored as an inexpensive and efficient way to collect data for data mining, scientific research, etc. [1,2]. Crowdsensing is a form of crowdsourcing in the IoT, where data collection is accomplished by distributing sensing tasks to ordinary users with sensing devices [3]. However, the data collected by this way is not completely reliable. The data submitted by workers is often noisy and the information may conflict with each other, due to factors such as the quality of the workers' sensing equipment and the dynamic movement of the equipment. How to find the most reliable truth among these conflicting information is an important challenge for crowdsensing [4,5].

One of the most popular ways to solving this problem is to perform truth discovery [6,7]. Truth discovery evaluates the trustworthiness of each source and determines the level of participation in data aggregation by trustworthiness. Various research works have given different frameworks for truth discovery. In this paper, we implemented one of the most widely used frameworks [6,7], we call it TruthFinder.

Data poisoning attacks against crowdsensing refers to attackers arrange a group of malicious workers to participate in sensing tasks, and submit some carefully designed malicious data, which can disrupt the effectiveness of data analysis [8]. By analyzing the trustworthness of data sources or task participants, it is possible to defend against various types of attacks to a certain extent [22–24], including data poisoning attacks. However, we focus on truth discovery. The current data poisoning attack methods against crowdsensing systems have reached a very sophisticated level, where attackers lurk in the platform for a long time to gain trust and gain a higher trustworthness for themselves, thus having the ability to launch more violent attacks [9,10]. Truth discovery algorithm does not take into account the presence of malicious workers. It is necessary to optimize and improve the TruthFinder framework to make it more robust to defend against data poisoning attacks.

In this paper, we focus on multi-round data poisoning attacks scenario and investigate how to improve the robustness of truth discovery. We validate the vulnerability of TruthFinder under multi-round data poisoning attacks, and we propose a high robust truth discovery with malicious worker detection (HRTDMD). The contributions of this paper are as follows.

1) We validate the vulnerability of the traditional TruthFinder, whice use three attack methods of multi-round data poisoning attack.
2) We make optimizations from two defense perspectives, malicious worker detection and improved TruthFinder. We propose a High Robust Truth Discovery with Malicious worker Detection (HRTDMD).
3) We normalize the criteria for the success of data poisoning attacks, and conduct experiments on real dataset, which fully demonstrate the effectiveness of our methods.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 shows three types of multi-round data poisoning attack meth-

ods and verifies the vulnerability of TruthFinder framework. Section 4 analyzes the way to defend against data poisoning attacks, and proposes HRTDMD. Section 5 illustrates the experimental results on real dataset. Section 6 concludes this paper.

## 2 Related Work

Recently, a lot of works focus on the problem of truth discovery. The method proposed by [11] focus on truth discovery in long-tail data. Truth discovery method in [19] removes bias from the data before processing. A truth discovery method to protect workers' privacy is proposed by [16]. Multi-truth discovery is studied in [20]. Some works studied the robustness of truth discovery, but they all focus on unintentional data quality issues, not data poisoning attacks. Different optimization methods around high robust truth discovery in classification tasks and numerical tasks are studied in [12–15,17]. Truth discovery by bootstrapping is introduced by [18], and truth confidence intervals can be discovered.

We focus on single truth discovery for numerical tasks. Truth discovery finds the truth from conflicting information by evaluating the trustworthiness of the data source. Our work is based on one of the most widely used frameworks TruthFinder [6,7]. The framework does not require additional prior knowledge, and iteratively computes and updates the possible truth and workers' trustworthiness. The current optimization for robustness of truth discovery is mainly in the following two ways.

- The first method of defense is to filter the data prior to data aggregation. Noise and malicious data can disrupt the normal operation of truth discovery. The method proposed by [17] is to perform source evaluation and source filtering prior to data aggregation. However, this work does not describe in detail the selection method of the evaluation threshold. The effectiveness of this defense method is directly dependent on the selection of the threshold.
- The second method of defense is how to discover the truth more accurately from the conflicting information. Bootstrapping method was used in [18] to sample the received data. Then perform truth discovery on each subset of the sample to obtain the truth set, and use the average of the truth set as the final truth. This method performs sampling for truth discovery, which introduces a significant time overhead, besides, this work focuses on confidence intervals for the truth, while our research work focuses on single truth.

Our work integrates source filtering and high robust truth discovery to defend against data poisoning attacks. We focus on the defense of multi-round data poisoning attack scenarios. There has been a significant amount of work done on single data poisoning attack, while more work is needed on multi-round data poisoning attack. Finally, we focus on numerical sensing tasks, which are more challenging compared to classification tasks.

# 3 Multi-round Data Poisoning Attacks Against TruthFinder

## 3.1 Multi-round Data Poisoning Attacks

In this paper, we consider a multi-round data poisoning attack scenario, where malicious workers continuously accept sensing tasks and submit carefully designed data over a period of time. In a multi-round attack scenario, we assume that a sensing task lasts for a period of time, and each worker receiving the task needs to submit data at each time step. The workers' weights (i.e., trustworthiness) are initialized at the initial moment of each task, and continuously adjust the weights according to the workers' performance in the task. The final weights obtained by the workers at the previous moment become the initial weights at current moment until the end of the task. We used the following three popular data poisoning attack methods to attack the defense framework.

**Hide-AttackPois:** In a task, the attacker adopts a method of alternate execution of hiding and attacking. Hiding means controlling the malicious workers to submit normal data to trick TruthFinder to assign a higher weight to them. When the weights are increased to a certain level, which means a higher probability of success, the attack action is selected and the malicious data is submitted to disturb the output. Please note that, under this method, even if malicious workers submit normal data, this data is still toxic. Malicious workers use it to disturb the weight assignment of TruthFinder. We mark $x : y$ after the method, where $x$ indicates the number of hiding and $y$ indicates the number of attacking. For example, $2 : 1$ means that in the whole process of attacking, the malicious workers alternately execute 2 hiding and 1 attacking.

**FullKnowledgePois:** We assume that the attacker has access to the weights assigned to the workers inside TruthFinder, and determines whether an attack should be launched based on the weights of the malicious workers. When the weight of the malicious worker is lower than the average value of the weights of the normal workers, the hidding action is taken, otherwise, the attacking action is launched.

**DeepPois:** This method was first proposed in [10], where the authors used deep reinforcement learning to model the data poisoning process, which is trained so that the model can automatically make decisions (attacking or hiding).

## 3.2 Vulnerability of TruthFinder

In this paper, we use TruthFinder, one of the most widely used truth discovery frameworks [6,7]. We assume that in a multi-round data collection task, TruthFinder will take the worker weights $\vec{W}^{(t-1)}$ at $t-1$ moments as the initial weights at $t$ moments, and iteratively update workers' weights $\vec{W}^{f}$ and aggregated values $\vec{X}^{*}$ until convergence or the preset maximum number of iterations $\eta$ is reached. Table 1 lists the key notation used in this paper.

**Table 1.** Summary of key notation.

| Notation | Definition |
|---|---|
| $N$ | Number of workers |
| $M$ | Number of task attributes |
| $\vec{X}$ | Set of worker's observations |
| $\vec{X}^*$ | Set of aggregated value during the iteration |
| $\vec{X}^f$ | Set of truth after truth discovery |
| $\vec{X}^g$ | Set of ground truth |
| $x_{n,m}$ | Observation of worker n for attribute m |
| $x_m^*$ | Aggregated value for attribute m during the iteration |
| $x_m^f$ | Aggregated value for attribute m after truth discovery |
| $x_m^g$ | Ground truth for attribute m |
| $\vec{W}$ | Set of worker's weights |
| $w_n^{(t)}$ | Weight of worker n at time t |
| $w_n^f$ | Weight of worker n during the iteration |

TruthFinder computes the truth $\vec{X}^f$ and woker's weights $\vec{W}$ by a two-step iterative formula as follows.

- Updated aggregated value $x_m^*$ by weighted average.

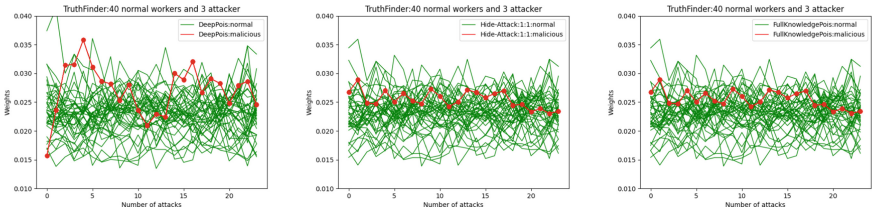
$$x_m^* = \frac{\sum_{i \leq N} w_i * x_{i,m}}{\sum_{i \leq N} w_i} \tag{1}$$

- The aggregated values $\vec{X}^*$ computed in the first step are used to further update the weights of workers $\vec{W}^f$. $d(x_{i,m}, x_m^*)$ denotes the distance (Euclidean distance, etc.) between the observation $x_{i,m}$ provided by worker $i$ and the aggregated value $x_m^*$ on the attribute m. The weights of the workers need to be normalized to ensure that $\sum_{n=1}^{N} w_n = 1$.

$$w_i = \sqrt{\frac{\sum_{m=1}^{M} \sum_{n \leq N} d(x_{n,m}, x_m^*)}{\sum_{m=1}^{M} d(x_{i,m}, x_m^*)}} \tag{2}$$

After the iteration stops, TruthFinder gives the final truth $\vec{X}^f$ (i.e., the truth closest to the ground truth) and the workers' final weights $\vec{W}$. We analyzed TruthFinder algorithm. TruthFinder does not take into account the long-term performance of workers, and workers' weights oscillate up and down in a wide range. A malicious worker, even if performed extremely poorly in the previous aggregation, can still trick TruthFinder into assigning a higher weight by submitting normal data in current aggregation. It is clearly unreasonable.

To validate the vulnerability of TruthFinder, we conducted an experiment using 7 months of real data, includeing a total of 214 tasks. The duration of




(a) DeepPois (b) Hide-AttackPois (c) FullKnowledgePois

**Fig. 1.** Weights of different attack method on TruthFinder (Color figure online)

each task was 24 h, and workers who accepted the task were required to submit observations at each hour. To avoid the coincidence of workers' weight changes in a single task, we averaged the changes in workers' weights over 214 tasks. The effect is shown in Fig. 1. The red line indicates the weight of malicious workers, and green lines indicates the weight of normal workers. As can be seen, all three attack methods can cheat a high weight in TruthFinder, which means they will have a large impact on the aggregation process of TruthFinder.

## 4 High Robust Truth Discovery with Malicious Worker Detection

### 4.1 High Robust Truth Discovery

TruthFinder uses a weighted average to calculate the truth, which is susceptible to extreme values. For practical reasons, the number of malicious workers planted in the mission by the attacker is limited, most of the workers involved in the task should be normal workers. Therefore, we used the weighted median to calculate the truth, which better reflects the real distribution of the data.

We optimized the way we inherit the weights of TruthFinder. We impose penalties on underperforming workers by weighting them lower. However, there is no need to reward workers for good performance (since it is the workers' responsibility to submit quality data). As the weights of poorly performing workers are reduced, the weights of other workers are appropriately increased when normalizing the weights. But the increase is within a small range and we do not perform any additional operations to increase a workers' weight. The workers' weight needs long-term good performance to maintain. The weight will be reduced once the worker has bad behavior, which can effectively mitigate the attacker's reliance on short-term disguise to deceive the trust of TruthFinder. Based on this idea, we propose a high robust truth discovery algorithm (HRTD). The specific calculations are shown below.

– Update the aggregated value $x_m^*$ by weighted median.

$$x_m^* = Median(w_1 * x_{1,m}, w_2 * x_{2,m}, ...w_i * x_{i,m}) \tag{3}$$

**Algorithm 1:** High robust truth discovery algorithm

**Input :** $\vec{X} = \{x_{1,1}, ..., x_{N,M}\}, \vec{W}^{(t-1)} = \{w_1, ..., w_N\}, N, M, \eta$
**Output:** $\vec{W}^{(t)}, \vec{X}^f$

$k \leftarrow 0$;
**while** *Not converage or* $k < \eta$ **do**
  $k \leftarrow k + 1$;
  *Update aggregate value* $\vec{X}^*$ *based on Eqn.* (3);
  *Update workers' weight* $\vec{W}^f$ *based on Eqn.* (4);
  *Normalized workers' weight* $\vec{W}^f$ *based on Eqn.* (5);
**end**
**for** $m \leftarrow 1$ **to** $M$ **do**
  $MAD_m = Median(abs(x_m^* - \vec{X}_m))$;
**end**
$\vec{Deviation} = \frac{\sum_{m=1}^{M} abs(x_m^* - \vec{X}_m) - MAD_m}{M}$;
$\vec{Rate} = abs(\frac{\sum_{m=1}^{M}(x_m^* - \vec{X}_m)}{M * x_m^*})$;
**for** $n \leftarrow 1$ **to** $N$ **do**
  **if** $Deviation_n > 0$ *or* $Rate_n > 0.05$ **then**
    $w_n^f = Min(w_n^f, w_n^{(t-1)})$;
  **end**
**end**
*Normalized workers' weight* $\vec{W}^f$ *based on Eqn.* (5);
$\vec{W}^{(t)} = \vec{W}^f$;
**return** $\vec{W}^{(t)}, \vec{X}^f$;

– Use the aggregated values $\vec{X}^*$ computed in the first step to update the weights of the workers $\vec{W}^f$.

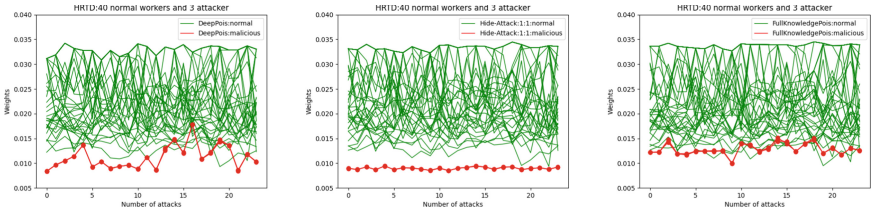
$$w_i = \sqrt{\sum_{m=1}^{M} \frac{|M| * \sum_{n=1}^{N}(x_m^* - x_{n,m})}{(x_m^* - x_{i,m})}} \tag{4}$$

– Normalize the workers' weights obtained in the second step.

$$w_i = \frac{w_i}{\sum_{n=1}^{N} w_n} \tag{5}$$

The purpose of the above calculations is to find the truth $\vec{X}^f$, but the workers' weights obtained here fluctuate a lot compared to the weights at the previous moment. We want to reduce this fluctuation as much as possible. We refer to the method of median absolute deviation (MAD), MAD is a robust measure of sample deviation for univariate numerical data. We take the MAD between the workers' observations $\vec{X}$ and the truth $\vec{X}^f$ as the bound. Deviations less than this bound are normal workers, while deviations greater than this bound are possible to be malicious and require further judgment. We further calculate

the relative error between the workers' observation $\vec{X}$ and the truth $\vec{X}^f$. If this error exceeds the threshold (We set the threshold to 0.05), which means that the workers' observation deviates far from the truth, the worker may be malicious. If the observation of a worker $n$ satisfies both of the above conditions, this worker $n$ needs to be penalized. The weight of this worker $n$ is chosen as the smallest of $w_n^{(t-1)}$ and $w_n^f$. If the worker does not satisfy the above condition, the workers' weight is $w_n^f$. Finally, we normalize the weights of the workers. The specific computations process can be seen in Algorithm 1.




(a) DeepPois (b) Hide-AttackPois (c) FullKnowledgePois

**Fig. 2.** Weights of different attack method on HRTD

To verify the effectiveness of our improvements, we use the three attack methods mentioned above to attack the high robust truth discovery (HRTD). The effect is shown in Fig. 2. As we can see, the weights of malicious workers are limited to a very low range under all three attack methods. This means that the role that malicious workers can play in data aggregation is very limited, proving that our optimization is effective. We further analyze the experimental results, the weights of malicious workers are effectively reduced, but still maintain about 0.01. Those weights will still have an impact. Therefore, we consider setting up a malicious detection in the platform to remove malicious workers in a timely manner.

### 4.2 Malicious Worker Detection

We adds a malicious worker detection step on HRTD to to remove malicious workers in a timely manner and get HRTDMD. The basic idea of malicious worker detection is shown by Fig. 3. The weights of workers in the same task are temporally correlated. When an attacker controls malicious workers, malicious workers tends to take the same action at the same moment (the overhead of training each malicious worker individually is significant), submitting the same malicious data or malicious data with appropriate noise added. Therefore, malicious workers tend to have similar weight trajectories. We use a clustering algorithm to detect malicious workers by inputting the weight trajectories of workers from the initial moment to the current moment into the model. We will detect at each hour and remove malicious workers once they are detected, and the weights

of the remaining normal workers need to be initialized. It is necessary to initialize the weights of normal workers because the behavior of malicious workers can disturb the weights of normal workers.

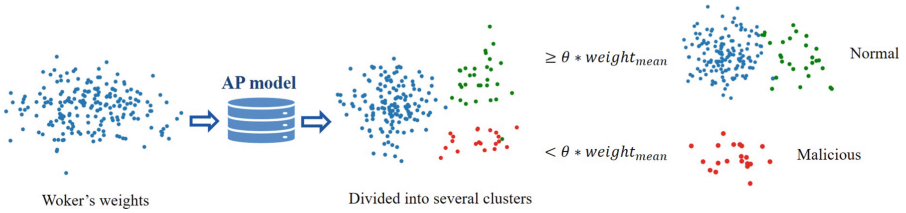



**Fig. 3.** Malicious worker detection with AP.

We use adaptive affinity propagation clustering (AP) [21]. AP algorithm is a deterministic clustering algorithm, and the clustering results are generally very stable. The algorithm does not need to define the number of clusters in advance, it keeps searching for suitable clustering centers during the iterative process. The clustering center can well represent the weight trajectory of the cluster. We select the weights of the last three time steps of each clustering center, and determine whether the mean value of the weights is less than $\theta * Mean(\vec{W})$. If so, the cluster is considered as malicious. In our experiments, we set $\theta = 0.6$.

## 5 Experimental Study

### 5.1 Experimental Setup

The experiments are run on a machine with Intel Xeon Gold 5118 CPU, GeForce RTX 2080 Ti graphics card and 32 GB of RAM. The real dataset used in the experiment consists of air quality data generated by 40 normal workers located in Krakow, Poland[1]. We used a total of 7 months of real data, containing a total of 214 tasks, each containing 24 rounds of data collection. Each data collection includes six attributes of air $\{temperature, humidity, pressure, pm1, pm25, pm10\}$. The important configuration informations in this experiment are as follows.

**Task:** Our experimental task is a 24-hour cycle and each worker submits observations every hour. After collecting the data, the truth discovery process is performed and the truth are published. In the same task, the weights of workers at $t$ moments are passed to $t + 1$ moments. TruthFinder iterates $\eta = 20$ times at most in each hour, and if convergence is not reached within 20 times, the aggregated value of the 20th iteration is used as the truth. The weights of the workers are reset every 24 h.

[1] https://airly.eu/.

**Baseline:** We call the approach in this paper HRTDMD, and compare it with the following three Baselines.

- TruthFinder [6]: The traditional TruthFinder, which uses a two-step iterative formula to compute truth and workers' weights.
- Cut-tail TruthFinder (CutTF): Cut-tail data is used for truth discovery and workers' weights estimation. Cut-tail data means removing the extreme values at both ends of the initial data and keeping the middle part. We use the weighted average of the remaining data as the truth, the weights are calculated based on Eq. (2). In this paper we used 20% of the data removed from each end.
- Bootstrapping TruthFinder (BootTF) [18]: The dataset is repeatedly sampled to obtain multiple samples, and for each sample, the TruthFinder is used to find the truth. Finally, the mean of the estimated truth of multiple samples is used as the final truth. The weight of each worker is calculated based on the final truth. In this paper, we repeated the sampling 10 times.

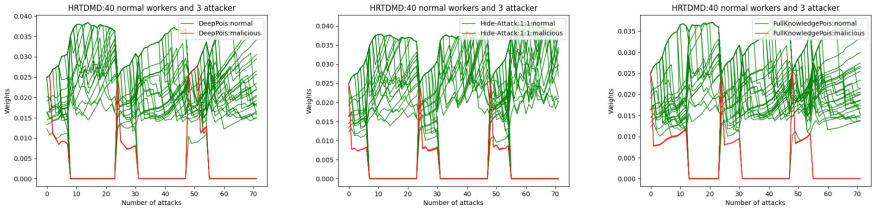
**Evaluation criteria:** In order to have a uniform criterion to evaluate the effectiveness of the defense, we quantify uniformly whether the attack is successful or not. We take the median of the normal worker observations as the ground truth $\vec{X}^g$. It is reasonable. Even if individual workers among normal workers submit data with noise for some reason, most of the workers' data are normal. We use the median to mitigate the interference of extreme values. We calculated the difference between the ground truth $\vec{X}^g$ and the truth $\vec{X}^f$ after launching the data poisoning attack based on Eq. (6). The attack is considered successful when the difference is greater than 0.05, otherwise it is considered a failure.

$$diff(\vec{X}_t^g, \vec{X}_t^f) = \left[\frac{2(x_{t,1}^f - x_{t,1}^g)}{|x_{t,1}^f| + |x_{t,1}^g|}, ..., \frac{2(x_{t,M}^f - x_{t,M}^g)}{|x_{t,M}^f| + |x_{t,M}^g|}\right] \tag{6}$$

### 5.2 Robustness of HRTDMD

We used three attack methods, DeepPois, Hide-AttackPois, and FullKnowledge-Pois, to attack the HRTDMD framework. We set the percentage of malicious workers to be about 7%, the number of normal workers is 40 and the number of malicious workers is 3. We used 214 different tasks for experiments, and since the moment when the platform detects malicious workers is not fixed, we cannot directly show the weight of 214 tasks on average as in the above paper. Therefore, we show the effect of 3 tasks so that we can clearly show the effect of our defense.

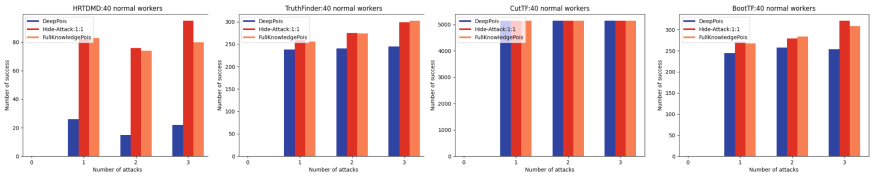
The effect is shown in Fig. 4. It can be seen that the weight of malicious workers are reduced to 0 under all three attack methods, which means they cannot play a role in the subsequent tasks, which also proves that our detection mechanism is effective. It can be seen that some of the normal workers are also detected as malicious workers by HRTDMD, these workers submit low-quality data due to equipment failure or some other reasons, it is reasonable to remove them.




(a) DeepPois (b) Hide-AttackPois (c) FullKnowledgePois

**Fig. 4.** Weights of different attack method on HRTDMD

We also conducted a comparison experiment to verify the high robustness of HRTDMD compared to the three Baselines. We use three attack methods to attack four defense methods. We set the percentage of malicious workers to be about 2.4%–7%, the number of normal workers is 40 and the number of malicious workers is 1, 2, 3. We counted the number of successful attacks for these 214 tasks.




(a) HRTDMD (b) TruthFinder (c) CutTF (d) BootTF

**Fig. 5.** Number of success on different defensive framework

As shown in Fig. 5, HRTDMD has the lowest number of successful attacks, which means that we have the best defense effect. CutTF has the highest number of successful attacks, which means the worst defense. This result is predictable, the direct tail-cutting operation on the data makes the remaining data center deviate in a certain direction. The truth found on this basis will also deviate farther from the ground truth. This also proves that it is necessary for us to perform truth discovery. The defense effect of BootTF is not significantly improved compared to TruthFinder. We analyzed that since only 10 repetitions of sampling were performed in the experiment, we could not sample each data evenly. However repeating the sampling 10 times brings a significant time overhead, which means that the truth discovery needs to be repeated 10 times. When in a larger data collection task, the overhead imposed by it would be very large.

## 6 Conclusions

In this paper, we focus on a truth discovery against multi-round data poisoning attacks in crowdsensing systems. We validate the vulnerability of TruthFinder

to data poisoning attacks. Then, we propose high robust truth discovery with malicious worker detection (HRTDMD) and validate its effectiveness on a real dataset. We also consider CutTF to demonstrate the necessity of performing truth discovery, and analyze the effect brought by Bootstrapping on TruthFinder. In the future, we will continue to study data poisoning attacks and defenses.

## References


1. Labrinidis, A., Jagadish, H.V.: Challenges and opportunities with big data. In: Proceedings of the VLDB Endowment (2012)
2. Estellés-Arolas, E., González-Ladrón-de-Guevara, F.: Towards an integrated crowdsourcing definition. J. Inf. Sci. **38**(2), 189–200 (2012)
3. An, J., Gui, X., Wang, Z., Yang, J., et al.: A crowdsourcing assignment model based on mobile crowd sensing in the Internet of Things. IEEE Internet Things J. **2**(5), 358–369 (2015)
4. Ganti, R.K., Ye, F., Lei, H.: Mobile crowdsensing: current state and future challenges. IEEE Commun. Mag. **49**(11), 32–39 (2011)
5. Capponi, A., Fiandrino, C., Kantarci, B., et al.: A survey on mobile crowdsensing systems: challenges, solutions, and opportunities. IEEE Commun. Surv. Tutorials **21**(3), 2419–2465 (2019)
6. Yin, X., Han, J., Philip, S.Y..: Truth discovery with multiple conflicting information providers on the web. In: TKDE (2008)
7. Li, Y., Gao, J., Meng, C., et al.: A survey on truth discovery. ACM Sigkdd Explor. Newslett. **17**(2), 1–16 (2016)
8. Miao, C., Li, Q., Xiao, H., et al.: Towards data poisoning attacks in crowd sensing systems. In: MobiHoc (2018)
9. Fang, M., Sun, M., Li, Q., et al.: Data poisoning attacks and defenses to crowdsourcing systems. In: WWW (2021)
10. Li, M., Sun, Y., Lu, H., et al.: Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems. IEEE Internet Things J. **7**(7), 6266–6278 (2019)
11. Li, Q., Li, Y., Gao, J., et al.: A confidence-aware approach for truth discovery on long-tail data. In: Proceedings of the VLDB Endowment (2014)
12. Li, Q., Li, Y., Gao, J., et al.: Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In: SIGMOD (2014)
13. Yin, X., Tan, W.: Semi-supervised truth discovery. In: WWW (2011)
14. Dong, X.L., Berti-Equille, L., Srivastava, D.: Integrating conflicting data: the role of source dependence. In: Proceedings of the VLDB Endowment (2009)
15. Zhang, D., Wang, D., Vance, N., et al.: On scalable and robust truth discovery in big data social media sensing applications. In: IEEE Transactions on Big Data (2018)
16. Xu, G., Li, H., Liu, S., et al.: Efficient and privacy-preserving truth discovery in mobile crowd sensing systems. In: IEEE Transactions on Vehicular Technology (2019)
17. Huang, Z., Pan, M., Gong, Y.: Robust truth discovery against data poisoning in mobile crowdsensing. In: GLOBECOM (2019)
18. Xiao, H., Gao, J., Li, Q., et al.: Towards confidence in the truth: a bootstrapping based truth discovery approach. In: SIGKDD (2016)

19. Li, Y., Sun, H., Wang, W.H.: Towards fair truth discovery from biased crowdsourced answers. In: SIGKDD (2020)
20. Lin, X., Chen, L.: Domain-aware multi-truth discovery from conflicting sources. In: Proceedings of the VLDB Endowment (2018)
21. Wang, K., Zhang, J., Li, D., Zhang, X., et al.: Adaptive affinity propagation clustering. ArXiv preprint arXiv (2008)
22. Tian, Z., Gao, X., Su, S., et al.: Vcash: a novel reputation framework for identifying denial of traffic service in internet of connected vehicles. IEEE Internet Things J. **7**(5), 3901–3909 (2019)
23. Tian, Z., Li, M., Qiu, M., et al.: Block-DEF: a secure digital evidence framework using blockchain. Inf. Sci. **491**, 151–165 (2019)
24. Sun, Y., Tian, Z., Li, M., et al.: Honeypot identification in softwarized industrial cyber-physical systems. IEEE Trans. Ind. Inf. **17**(8), 5542–5551 (2020)

# BERT-Based Vulnerability Type Identification with Effective Program Representation

Chenguang Zhu[1,2], Gewangzi Du[1,2], Tongshuai Wu[1,2], Ningning Cui[1,2], Liwei Chen[1,2(✉)], and Gang Shi[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{zhuchenguang,dugewangzi,wutongshuai,cuiningning,chenliwei,shigang}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

**Abstract.** Detecting vulnerabilities is essential to maintaining software security. At present, vulnerability detection based on deep learning has achieved remarkable results. The type of vulnerability could tell the vulnerability principles and help the programmer quickly pinpoint the precise location of vulnerabilities. Moreover, the type of vulnerability is very valuable for remediating it. Therefore, it is essential to identify vulnerability types. This paper proposes a new vulnerability type identification framework based on deep learning. The framework is based on syntax and semantics, and the detection granularity is fine to the slice level. To include comprehensive vulnerability types, we use four slicing methods to represent the program. In addition, we model four kinds of code slice features based on BERT. For evaluation, we used 64 three-level CWE-IDs vulnerability types in National Vulnerability Database (NVD) and Software Assurance Reference Dataset (SARD) for vulnerability type identification. The experimental results show that it has significant performance in vulnerability type identification.

**Keywords:** Vulnerability detection · Type identification · Bidirectional self-attention mechanism

## 1 Introduction

As pressure on enterprises to swiftly identify software vulnerabilities has increased, security analysts have adopted automated vulnerability detection methods. Once a vulnerability is detected, the type of vulnerability is very valuable for remediating it. Unfortunately, modern software systems are extremely complicated. For example, a typical product level software has million level codes, and contains a security vulnerability for every 1000 lines of code [1]. Detecting and identifying the type of vulnerability in such a great amount of codes is extremely hard.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 271–282, 2022.
https://doi.org/10.1007/978-3-031-19208-1_23

Existing efforts show the effectiveness of combining program analysis with deep learning techniques to address the vulnerability detection challenges [2]. The deep learning-based methods relieve human experts from the arduous task of defining vulnerability characteristics. Compared to traditional machine learning techniques, deep learning can automatically learn more complex and abstract high-level features or representations to capture code semantics [3].

However, most of the current deep learning-based vulnerability detection work is binary classification [4], which means that they can only output whether a code segment is vulnerable. The type of vulnerability could tell the vulnerability principles and help the programmer quickly pinpoint the precise location of vulnerabilities [5]. Therefore, it is of great significance to design deep learning vulnerability detection to classify the vulnerabilities types accurately.

There are two challenges when identifying the types of vulnerabilities using deep learning-based vulnerability detection methods. First, it is difficult to obtain meaningful representations of programs related to vulnerability types. Existing methods learned Natural Language Processing (NLP) models on code's raw text. However, the underlying structure of the language (i.e., tree structures) and its semantic, which is valuable for identifying the vulnerability type, is hard to extract from raw text data. Second, to accurately identify vulnerability types, the deep learning model needs to precisely understand the semantic information of the code.

To solve these two problems, we propose a novel vulnerability type identification framework. For the first time, we leverage four kinds of syntax features to represent the code in vulnerability type identification, and model them using the bidirectional self-attention mechanism to encode code structure and semantics for detecting vulnerability types. We summarize three contributions of our work as follows:

1) **A vulnerability type detection method based on four kinds of syntax features.** We take code slicing as granularity, and this paper adopts four kinds of slices methods: API function, arithmetic expression, pointer usage, and array usage. Compared with the code fragment of only one kind of slice (API function) [6], four kinds of slices can more comprehensively contain the specific types of vulnerabilities.
2) **Bidirectional self-attention mechanism.** We use the bidirectional multi-head self-attention mechanism to represent code slice for the first time. It can enhance the semantics of the code from the context information around the token and understand the semantics of the token in different semantic scenarios, which is helpful to detect the types of vulnerabilities.
3) **Vulnerability type identification.** We propose a vulnerability type identification prototype, which detects the types of vulnerabilities in NVD and SARD. The evaluation shows that our method can accurately detect the vulnerability types.

**Paper Organization**. Section 2 describes the related work. Section 3 discusses the framework of the vulnerability type identification method. Section 4 dis-

cusses the design of this method. Section 5 introduces our experimental results. Section 6 concludes the paper and prospects for future work.

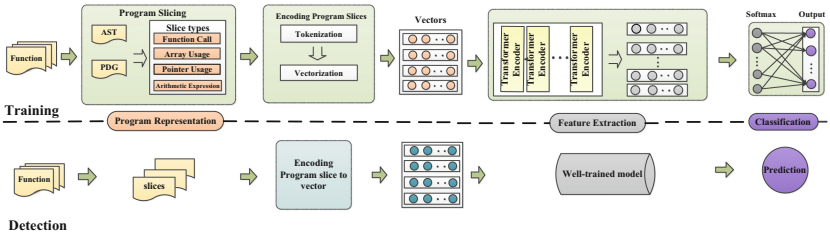



**Fig. 1.** Overview of framework.

## 2 Related Work

A recent broad development in vulnerability detection is the integration of deep learning techniques [1]. Guanjun Lin et al. proposed a cross-project scenario approach [7], which detects vulnerabilities on target projects by learning transferable representations from vulnerability data of historical software projects. On this basis, they also proposed the first cross-domain software [8]. They use the Metric Transfer Learning Framework algorithm to learn transformation matrices that map high-level representations to cross-domain representations. SySeVR [9] represents programs as vectors and preserves the syntactic and semantic information of the code that reflects vulnerability characteristics. Deqing Zou et al. proposed the milestone method Vuldeepecker [5], which uses a program representation called "code gadget" to detect buffer errors and resource management errors (unfortunately, "code gadget" can only be performed using commercial tools extraction). However, none of the above methods can detect the type of vulnerability. As far as we know, at present, there are few studies on vulnerability type identification based on deep learning. $\mu$Vuldeepeeker is a deep learning vulnerability identification [6], which the code sequence forms a context to capture the global semantics related to possible vulnerabilities. It only uses one code slice (API function) and takes BLSTM as a deep learning algorithm. However, taking sensitive API functions as the point of interest, the slices obtained by this method may miss some features of vulnerability types, making the program characterization unable to cover all vulnerability types. At the same time, to more accurately identify the types of vulnerabilities, we need to better understand the program semantics when modeling the program representation. To sum up, the information of vulnerability types is helpful to locate vulnerabilities accurately, so vulnerability type identification needs further research.

# 3 Framework

## 3.1 Basic Idea

API/library function call(FC), arithmetic expression(AE), usage of the pointer (PU), and usage of the array (AU) are the critical information to judge the type of vulnerability. In addition, through observation, we found that this function slicing method based only on API calls cannot cover all vulnerability types. Taking the dataset of this paper as an example, it contains 126 types of CWE-ID vulnerability. The API/library function call slice can cover 106 types of CWE-ID vulnerability. Only when we use four kinds of slicing methods can slicing cover all function vulnerability types, as shown in Table 1.

**Table 1.** Number of CWE-IDs covered by code slices

| Code slice type | $FC$ | $FC \cup AE \cup PU$ | $FC \cup AU \cup PU$ | $FC \cup AE \cup AU$ | $ALL$ |
|---|---|---|---|---|---|
| Cover CWE-ID | 106 | 116 | 123 | 125 | 126 |

Therefore, we use four kinds of statement-level slices to represent this information in the code, rather than simply feeding the whole code to the model. Then we should model the syntax structure and semantic using these four kinds of slices.

Because the vulnerability type information is closely related to the semantic information of code fragments, the semantics of the same token in diverse code contexts are different, which is affected by its control dependency and data dependency. To accurately understand the semantics of the code, we propose a code self-attention mechanism to calculate the relationship between each token and other tokens in the code to enhance the semantics of the token.

## 3.2 Framework Overview

We slice the source code program, generate its Abstract Syntax Trees(AST), Control Flow Graph(CFG), Program Dependency Graph (PDG), and generate slices by matching four syntax types, and we take this fragment as input. In this paper, this model converts each token into a one-dimensional vector by querying the word vector. The embedding method is WordPiece [10]. We calculate the multi-head attention layer to obtain the high-level features of code fragments. We use a Softmax layer to represent high-level features as a probability distribution and finally get a classification result. Finally, we use NVD and SARD datasets to detect specific types of vulnerabilities under this framework, as shown in Fig. 1.

# 4 Design

## 4.1 Program Slicing

From our observation, it is more accurate to detect the specific types of vulnerabilities at the code slice level than at the function level. Therefore, we choose a fine-grained code slice as the detection unit. To contain more key syntax information, we use four syntax types for slicing.

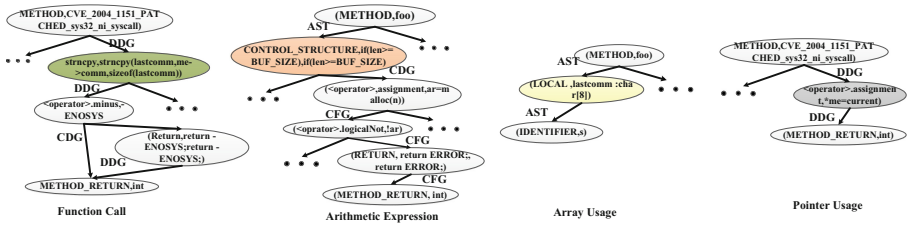



**Fig. 2.** Four kinds of syntax features in AST.

**Step1**: Generate AST. We use Joern to generate the AST [11] of the functions in the code fragment. The AST is a tree-structured representation of code nodes and types, the syntax features of the four types are shown in Fig. 2.

**Step2**: Choose the interest point. Selecting the syntax and semantic features of code is very important for code slicing. In Fig. 2, the nodes marked with color are the interest points. The start line of code slice is a statement that contains the interest point.

**Step3**: Generate the PDG. The CFG analyzes all the paths the program takes during its execution. The vertex of PDG the same as in CFG. The edges of PDG can represent a data or control dependency between a pair of nodes, which induce the code semantic.

**Step4**: Code Slicing. We extracted statements which can be reached the interest point via PDG. Figure 3 shows examples of four kinds of slices. The slice are inter-procedural via function calls. We consider forward and backward slices. From the start line, we perform the inter-procedural backward slice based on control-dependency and data-dependency. Because statements affected by the interest point via control dependency will not be vulnerable in almost cases, we perform the inter-procedural forward slice based on only data-dependency. A code slice can be extracted from a interest point. For example, in Fig. 3(a), the interest point of API/Library Function slicing is *memcpy*, and the starting line of slicing is key line 9 (red mark).

**Step5**: Label Code Slices. Our dataset is manually audited for vulnerability locations based on vulnerability descriptions by professional software security researchers. We label code snippet by determining if the code slice contains the vulnerability location and CWE-ID. 0 means no vulnerability, $i$ is the specific vulnerability type.

```
CWE-200 CVE_2010_4075.c
      memcpy 9
1 static int CVE_2010_4075_
      ...
4 struct serial_icounter_struct
      __user *icnt)
5 struct uart_icount cnow ;
6 struct uart_port * uport =
      state -> uart_port ;
9 memcpy ( & cnow , & uport
      -> icount , sizeof ( ... ) );
   ...
24 return copy_to_user ( ... )
      ...: 0 ;
```

(a) API/Library Function Call

```
CWE-199 CVE_2011_0021.c
      int dx = i_shifth + x ; 52
1 static int
      CVE_2011_0021_VULN_
      ( ... )
6 int i_shifth ;
7 int i_shiftv ;
8 int x , y ;
   ...
50 for( x = 0; x <
      CDG_SCREEN_WIDTH;
      x++ )
52 int dx = i_shifth + x ;
56 dy = ( dy + ...
   ...
```

(b) Arithmetic Expression

```
CWE134_Uncontrolled...17.c
      data 50
34 void CWE134_...17_bad()
36 int i , j ;
37 char * data ;
38 char dataBuffer [ 100 ] = ""
      ;
39 data = dataBuffer;
   ...
50 strncat ( data + dataLen ,
      ... );
57 fprintf ( stdout , data );
```

(c) Pointer Usage

```
CWE127_Buffer...33.cpp
      dataBuffer 34
26 void bad()
28 wchar_t * data ;
30 wchar_t dataBuffer [ 100 ] ;
31 wmemset ( dataBuffer ,...
32 dataBuffer [ 100 - 1 ] = L
      '\0'
34 data = dataBuffer - 8;
42 wcscpy ( dest , data );
43 printWLine ( dest );
18 void printWLine (const ...)
...
```

(d) Array Usage

**Fig. 3.** Four kinds of code slices (Color figure online)

### 4.2 Encoding

**Tokenization.** After slicing the code, we get sliced fragments in four kinds of code slicing syntax. Then, we vectorize the code snippets as the input for the model. According to the heuristic, the function and variable names in the token are the key information of the vulnerability type, and they are the advanced features of the vulnerability type that the deep neural network needs to extract.

In deep learning based vulnerability type identification, we use the uniform name style to symbolize function and variable names. For example, function names are represented as $Function0$, $Function1$, ..., $FunctionN$, variable names are represented as $Variable0$, $Variable1$, ..., $VariableN$. At the same time, we also remove non-ASCII characters.

**Vectorization.** We use the WordPiece tool to get the tag vocabulary by word segmentation with the source code. Compared with Word2Vec tool, this embedding method can divide words into finer word segments.

### 4.3 Feature Extraction

Compared with RNNs, BERT-Based model can perform concurrent execution, extract the relational features of words in sentences, and extract relational features at multiple levels, thereby more comprehensively reflecting sentence semantics [12]. The reason we use BERT-Based is as follows. **1)Pre-training model.** The BERT model is one of the most effective pre-training models, which performs well in text classification tasks. Through fine-tuning, the model trained in our dataset can be used by other datasets which include new vulnerability types [12]. **2) Bidirectional self-attention.** Vulnerability type identification requires an accurate understanding of the semantics of the code. BERT contains multiple transformer modules, transformed into a multi-layer bidirectional transformer, using bidirectional self-attention. The multi-layer bidirectional self-attention mechanism will be explained in detail.

```
1 int foo(char *str, size_t n)
2 {
3     char buf [BUF_SIZE], *ar;
4     size_t len1=strlen(str);
5     if (len1>=BUF_SIZE) return ERROR;
6     memcpy(buf, str, len1)
7     ar=malloc(n);
8     if (!ar) return ERROR;
9 }
```

(a) Non-vulnerable Program

```
1 int foo(char *str, size_t n)
2 {
3     char buf [BUF_SIZE], *ar;
4     size_t len2=strlen(str);
5     if (len2>=2*BUF_SIZE) return ERROR;
6     memcpy(buf, str, len2)
7     ar = malloc(n);
8     if(!ar) return ERROR;
9 }
```

(b) CWE-119 Vulnerable Program

**Fig. 4.** Examples of semantic understanding in different contexts

The semantics of the same token is different in diverse sentences, and this situation can be a key feature of vulnerability types [13]. For example, as shown in the Fig. 4, non vulnerability code fragment and CWE-119 code fragment [14]. We observed that in line 6, the operation of the parameter *len* of the *memcpy* function determines the vulnerability or not and the vulnerability type. In Fig. 4(a), in the judgment statement in line 5, $len1 \geq BUF\_SIZE$ can avoid buffer overflow, so this statement is a code line without loopholes; however, the *memcpy* function in Fig. 4(b), and in the judgment statement, $len2 \geq 2 * BUF\_SIZE$ may overflow the buffer, resulting in an error. Therefore, whether the operation of the same token (i.e. *memcpy*) is correct is related to the *len* parameter (below *memcpy*) and its judgment statement (above *memcpy*).

This paper introduces a bidirectional multi-head self-attention mechanism to address this situation. As shown in Fig. 5. The bidirectional multi-head attention mechanism can better understand the semantics of each tag of the code sequence in different semantic scenarios and combine with the left and right context information. At the same time, the multi-head bidirectional self-attention mechanism can solve the limitation of long-distance interdependence of RNNs.

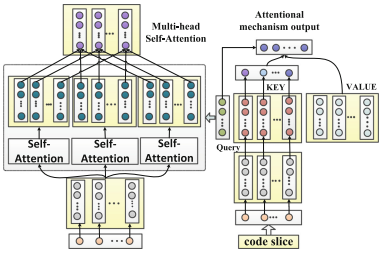



**Fig. 5.** Bidirectional self-attention mechanism.

### 4.4 Classification

This Softmax layer represents high-level features. Use the softmax function to convert the slice vector to a probability distribution. Then the model continuously updates the parameters to minimize the categorical cross-entropy loss function.

## 5 Experiments and Results

We raised the following three Research Questions (RQs) and answered them through our experiments.

**RQ1**: Does four kinds of code slicing effective in vulnerability type identification?

**RQ2**: Is the bidirectional self-attention model based on four code slice types efficient in vulnerability type identification?

**RQ3**: Does the bidirectional self-attention model based on four types of code slices effective in vulnerability detection?

The operating system of the server is Ubuntu Linux 18.04, with NVIDIA GeForce RTX 2080Ti GPU and Intel(R) Xeon(R) Silver 4214 CPU @ 2.20GHz.

### 5.1 Evaluation Metrics

We use the vulnerability type evaluation index in the vulnerability type identify task: M_FPR, M_FNR, M_F1, W_FPR, W_FNR, and W_F1. The $M\_*$ is the vulnerability type indicator. The $W\_*$ metric is the weight of each vulnerability type.

### 5.2 Dataset and Experimental Setup

For NVD and SARD, we used 15,591 C/C++ programs. Vulnerable programs account for about 95% of the total. The vulnerability sources are SARD and NVD. We have generated 420,627 code slices by four kinds of slice types, including 56,395 slices with vulnerable statements and 364,232 slices without vulnerabilities.

Our experiment uses random sampling to divide the data set. Experimental data set: validation set: test set = 8:1:1. Vocabulary construction: Segment the words of the code snippet, and collect the vocabulary to form the vocabulary corpus. Data set segmentation: process sliced data, delete spaces, clauses, line breaks, etc., and combine their labels with sliced data. During data embedding, to embed longer code sequences, the maximum length of the vector is set to 1000. The learning rate and batch size in the training process are set to $2^{e-5}$ and 6, respectively; epochs are 5. We choose categorical cross-entropy as loss function.

### 5.3 Label

Since CWE-IDs are hierarchical, we use the third-level of the CWE-ID tree as the vulnerability type, so the CWE is aggregated to the third-level CWE-ID as the label. For example, CWE-666 is a sub-type of CWE-415, and CWE-415 is at the third-level of the CWE-ID tree, so CWE-415 is also used as the vulnerability type for the CWE-666 vulnerability. We summarized 126 CWE-IDs into 64 third-level CWE-IDs. We labeled code slices: 0 means no vulnerability, i is the specific vulnerability type($1 \leq i \leq 64$).

### 5.4 Experimental Results

**Experiments for Answering RQ1.** Table 2 summarizes the experimental results. We used only FC slices and four kinds of code slices for the program representation, and compared their representation effects. Through our observation, when we use four kinds of slices(i.e., ALL slices) to represent vulnerability type

information, the evaluation index based on BERT and RNNs (BGRU, BLSTM) modeling has been improved, as shown in the Fig. 6(a).

Where M_FPR decreased by 0.06% and W_FPR decreased by 1.35% on. Specifically, M_FNR decreased by 9.14% and W_FNR decreased by 4.76% on average, the situation of false negatives has been significantly improved. F1 score was improved, M_F1 and W_F1 respectively increased by 4.63% and 9.14% on average.

**Table 2.** Effects of different models in vulnerability identification.

| Slice type | Model | M_FPR | M_ FNR | M_ F1 | W_ FPR | W_FNR | W_F1 |
|---|---|---|---|---|---|---|---|
| FC | RNNs-BLSTM | 0.11% | 25.11% | 83.30% | 1.91% | 11.00% | 81.11% |
| | RNNs-BGRU | 0.10% | 24.71% | 83.10% | 1.89% | 10.60% | 81.49% |
| | BERT-Based | 0.05% | 18.41% | 87.46% | 1.56% | 9.37% | 89.60% |
| ALL | RNNs-BLSTM | 0.04% | 17.67% | 86.50% | 0.53% | 7.39% | 90.51% |
| | RNNs-BGRU | 0.04% | 16.48% | 87.10% | 0.47% | 7.17% | 89.60% |
| | **BERT-Based** | **0.01%** | **5.53%** | **94.16%** | **0.31%** | **2.16%** | **97.50%** |

**Insight 1:** The four kinds of code slice representation used in this paper can enrich the information of vulnerability types. This code slicing method can obtain enough vulnerability-type information.

**Experiments for answering RQ2.** When we use bidirectional self-attention mechanism to model four kinds of code slices, compared with RNNs, BERT-Based's M_FPR was 0.03% lower, W_FPR decreased 0.19%, M_FNR decreased 12.11%, W_FNR decreased by 5.12%, M_F1 increased by 7.36%, and W_F1 increased by 6.45% on average. As shown in Table 2 and Fig. 6(b). In particular, W_F1 is 97.50%, indicating that our method significantly affects vulnerability type detection based on category weight.

**Insight 2:** BERT-Based performs well in vulnerability type identification, and the detection effect of the bidirectional self-attention mechanism model is better than RNNs. It shows that the bidirectional self-attention mechanism can accurately capture the semantic information of the code slice, to characterize the key information of the vulnerability type.


(a) ALL vs FC (b) BERT-Based vs RNNs

**Fig. 6.** Comparison BERT-Based with other methods.

**Experiments for Answering RQ3.** We compare our most effective model Bert-based with the open-source static analysis tools Flawfinder and RATS, and we also compare it with an effective deep learning detector VulDeePecker [15] and SySeVR [9]. Table 3 summarizes the experimental results. We observe that the results of BERT-Based are significantly improved. Compared with SySeVR, Acc and FPR are almost the same as SySeVR. Pre is 0.4% higher, F1 is 0.7% higher, and FNR is 1.2% lower. Note that the results of other models are from [9].

**Table 3.** Effects of different models in vulnerability detection.

| Method | Acc | Pre | F1 | FPR | FNR |
|---|---|---|---|---|---|
| Flawfinder | 69.8% | 22.8% | 25.7% | 21.6% | 70.4% |
| RATS | 67.2% | 12.8% | 13.7% | 21.5% | 85.3% |
| VulDeePecker [15] | 92.2% | 78.0% | 66.6% | 2.5% | 41.8 % |
| SySeVR [9] | 98.0% | 90.8% | 92.6% | 1.4% | 5.6% |
| **BERT-Based** | **98.1%** | **91.2%** | **93.3%** | **1.5%** | **4.4%** |

**Insight 3:** The results show that the effect of BERT-Based in vulnerability detection is better than the SySeVR model, which shows that the representation based on bidirectional self-attention mechanism can better capture information to detect vulnerabilities accurately.

## 6 Conclusion

When detecting vulnerability type identification, we use the slicing method of four AST syntax types, which can contain comprehensive vulnerability types. The comparison between experiments and other studies further proves that this slicing method can provide complete key code fragment information and enrich the knowledge of vulnerability types. In addition, in the slice-level code, the bidirectional multi-head self-attention mechanism is used to detect the type of vulnerability for the first time. The model framework can consider the context information at the sentence level and the left and right knowledge of tokens at the toke level to help detect specific types of vulnerabilities. In future research work, we will continue to study the detection of specific vulnerability types. In addition, vulnerability types can help locate vulnerabilities, worthy of further research.

**Acknowledgements.** This work is partially supported by the National Natural Science Foundation of China (No. 62172407), and the Youth Innovation Promotion Association CAS.

## References


1. Lin, G., Wen, S., Han, Q.L., Zhang, J., Xiang, Y.: Software vulnerability detection using deep neural networks: a survey. In: Proceedings of the IEEE, vol. 99, pp. 1–24 (2020)
2. Ghaffarian, S.M., Shahriari, H.R.: Software vulnerability analysis and discovery using machine-learning and data-mining techniques. ACM Comput. Surv. (CSUR) **50**(4), 1–36 (2017)
3. Yamaguchi, F., Wressnegger, C., Gascon, H., Rieck, K.: Chucky: exposing missing checks in source code for vulnerability discovery. In: ACM Conference on Computer and Communications Security (2013)
4. Grieco, G., Grinblat, G.L., Uzal, L., Rawat, S., Feist, J., Mounier, L.: Toward large-scale vulnerability discovery using machine learning. In: ACM (2016)
5. Zhen, L., Zou, D., Xu, S., Ou, X., Zhong, Y.: Vuldeepecker: a deep learning-based system for vulnerability detection. In: Network and Distributed System Security Symposium (2018)
6. Zou, D., Wang, S., Xu, S., Li, Z., Jin, H.: $\mu$VulDeePecker: a deep learning-based system for multiclass vulnerability detection. arXiv e-prints (2020)
7. Lin, G., Zhang, J., Wei, J., Lei, L., Yang, P., Xiang, Y.: Cross-project transfer representation learning for vulnerable function discovery. IEEE Trans. Indust. Inform. **14**, 3289–3297 (2018)
8. Liu, S., et al.: CD-VULD: Cross-domain vulnerability discovery based on deep domain adaptation. IEEE Trans. Depend. Sec. Comput. vol. 99, p. 1 (2020)
9. Li, Z., Zou, D., Xu, S., Jin, H., Chen, Z.: SySeVR: a framework for using deep learning to detect software vulnerabilities. IEEE Trans. Depend. Sec. Comput. **99**, 1 (2021)
10. Schuster, M., Nakajima, K.: Japanese and Korean voice search. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2012)
11. Yamaguchi, F., Golde, N., Arp, D., Rieck, K.: Modeling and discovering vulnerabilities with code property graphs. In: IEEE Symposium on Security and Privacy (2014)
12. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (2018)
13. Li, X., Gao, W., Feng, S., Zhang, Y., Wang, D.: Boundary detection with BERT for span-level emotion cause analysis. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021 (2021)
14. Padmanabhuni, B.M., Tan, H.: Buffer overflow vulnerability prediction from $\times$86 executables using static analysis and machine learning. In: Computer Software and Applications Conference 2015, pp. 450–459 (2015)
15. Li, Z., et al.: VulDeePecker: a deep learning-based system for vulnerability detection. arXiv preprint arXiv:1801.01681 (2018)

# Privacy-Preserving and Truthful Auction for Task Assignment in Outsourced Cloud Environments

Xufeng Jiang[1,2] and Lu Li[2,3](✉)

[1] Nanjing Tech University, Nanjing, China
[2] Yancheng Teachers University, Yancheng, China
lil@yctu.edu.cn
[3] University of Science and Technology of China, Suzhou, China

**Abstract.** Due to high fairness and allocation efficiency, the task assignment problem of mobile applications via auctions has become a promising approach to motivate bidders to provide their mobile device resources effectively. However, most of existing works focus on the auction mechanism under the plaintexts, and ignore the problems caused by information leakage. In this paper, we study the problem of the privacy-preserving auction for task assignment in outsourced cloud environments without leaking any private information to anyone. Specifically, we use Yao's garbled circuits and homomorphic encryption system as underlying tools. Along with several elaborately designed secure arithmetic subroutines, we propose a privacy-preserving and truthful auction framework for task assignment in outsourced cloud environments. Theoretically, we analyze the complexity of our scheme in detail and prove the security in the presence of semi-honest adversaries. Finally, we evaluate the performance and feasibility of our scheme through a large number of simulation experiments.

**Keywords:** Privacy-preserving · Auction · Task assignment · Yao's garbled circuits

## 1 Introduction

With the development of mobile applications, a single mobile device can no longer meet the resource requirements of mobile application tasks. On the one hand, due to expensive mobile devices, enterprises that need devices are reluctant to buy large amounts of devices to support the development or operation of mobile applications. On the other hand, it is unrealistic for manufacturers with idle mobile devices to share their device resources or perform tasks for others. In recent years, due to high fairness and allocation efficiency, cloud auctions for task assignment have become a promising approach to motivate bidders to provide their mobile device resources effectively. However, most of existing works [1–4] focus on the truthfulness, personal rationality and computational efficiency


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 283–295, 2022.
https://doi.org/10.1007/978-3-031-19208-1_24

of auctions, but ignore the security problems caused by information leakage in outsourced cloud environments. For example, bidders may eavesdrop on other bidders' bid information to modify their actual bids, which will win the auction with a higher probability; the cloud auctioneer may know the bidder's identity and bid information, which will tamper with the pricing strategy to obtain additional profits. The above problems will break the truthfulness and fairness of the auction. Therefore, sensitive data should be encrypted before uploading to the cloud auctioneer, who requires to perform the auction process on the encrypted data and output the same auction results as the original auction mechanism without leaking any intermediate results to anyone. In addition, it is also necessary to ensure the high efficiency of the system when dealing with large amounts of users in real-life applications. The above requirements make the privacy-preserving auction for task assignment a challenging task.

We focus on the problem of a privacy-preserving auction framework for task assignment in outsourced cloud environments. In this paper, we select the recent work [1] as the underlying auction mechanism. There are two reasons for our choice. First, this work designs an optimal winning bids determination algorithm and employs a one-to-many matching manner. However, the other works [2,3] limit the auction mechanism in a one-to-one matching manner, which omits the fact that the resource-rich devices can support the resource requirements of multiple buyers in a practical system. Second, this scheme has proven the properties of truthfulness, individual rationality, and system efficiency. Some works like [4] do not provide complete proof for these properties and have high system overhead. Recently, there exist lots of privacy-preserving solutions to tackle various cloud auction problems [5–13], which are introduced in Related Work. Nevertheless, none of the above works can directly deal with our problem. Specifically, since our auction process involves a lot of nonlinear arithmetic operations, which is hard to guarantee security throughout the whole auction process. For example, Jiang *et al.* [7] propose a secure auction scheme for task assignment, but this scheme do not consider the privacy of the number of resources required, which is critical data during the auction process. Wang *et al.* [11] propose a secure and truthful double auction scheme for heterogeneous spectrum allocation, but this scheme discloses the number of candidates, which leaks the privacy of data access patterns. In addition to security, ensuring the system efficiency of our privacy-preserving auction is still a challenging task. These recent works [5,8,9,12] design a serious of secure auction schemes to provide a strong security guarantee for bidders. However, these schemes involve large amounts of public-key encryption operations, which leads to huge computation and communication costs.

In this paper, we propose a Privacy-preserving and Truthful Auction scheme for Task Assignment (PTATA) based on a novel composite method of combining Paillier homomorphic cryptosystem [16] with Yao's garbled circuits [17], which fully protects the privacy information for each participant in the presence of semi-honest adversaries. Our contributions are as follows:

1. Based on Paillier homomorphic cryptosystem and Yao's garbled circuits, we propose a privacy-preserving and truthful auction scheme for task assignment

in outsourced cloud environments without leaking any actual intermediate results to anyone.
2. We design two secure arithmetic subroutines over the encrypted data, which can be the critical building blocks in other applications.
3. We prove that our scheme can guarantee a strong security under the semi-honest model and analyze the system complexity. Based on extensive experiments, we evaluate the performance and feasibility of our scheme.

The rest of our paper is organized as follows. In Sect. 2, we present problem formulation and primitives. Our scheme PTATA is presented in Sect. 3. In Sect. 4, we present our simulation experiments. Related works are discussed in Sect. 5. Finally, the conclusion is made in Sect. 6.

## 2 Problem Formulation and Primitives

### 2.1 System Framework

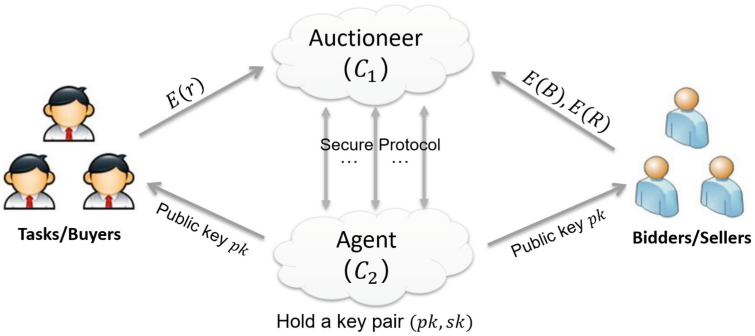



**Fig. 1.** System framework

As shown in Fig. 1, we construct the system framework of our problem under the semi-honest model [14]. Specifically, *Buyers* (or task demanders) submit the encrypted resource requirements and identity information to the *Cloud Auctioneer* ($C_1$), who performs the auction process over the encrypted data. *Sellers* (or bidders) have idle mobile devices and bid for each task. They encrypt these bids and the number of resources provided before uploading to the cloud auctioneer. The *Cloud Agent* ($C_2$) provides cryptographic services and helps the cloud auctioneer to execute the secure auction protocol. Note that the cloud auctioneer and cloud agent are competitive companies, that are highly improbable to conspire with each other, such as *Amazon* and *Google* [15]. Such a system framework is widely used in various related domains [5–9].

The main goals of our privacy-preserving auction scheme are as follows:

- **Correctness:** The results of the privacy-preserving auction scheme are consistent with the original auction mechanism.
- **Security:** Except for the auction results, all participants will not learn anything about the actual information during the auction process.
- **Efficiency:** In practical applications, it is important to ensure system efficiency when dealing with large users.

### 2.2 Auction Mechanism

In this paper, we consider a truthful auction for task assignment, where $n$ sellers want to compete for homogeneous tasks of $m$ buyers. Let $t_j(1 \leq j \leq m)$ denote the set of buyers, each of whom has one task that requires the same amount of resources required $r$. Let $d_i(1 \leq i \leq n)$ denote the set of sellers, each of whom provides a certain number of resources $R_i$ and bids $b_{i,j}$ for each task. Then, each buyer $t_j$ submits $r$ to the auctioneer while each seller $d_i$ submits $(B_i, R_i)$, where $B_i \in B$ is the set of bids by $d_i$. Note that each seller can meet the resource requirements of multiple buyers.

We review a truthful auction mechanism for task assignment [1]. The following is a brief description of this scheme.

**Step1: Winning Bids Determination.** For each seller $d_i$, the auctioneer first calculates the number of tasks that $d_i$ can accept, which is constrained as follows:

$$K_i = min\{\lfloor \frac{R_i}{r} \rfloor, m\}, (1 \leq i \leq n). \tag{1}$$

After obtaining $K = \{K_i\}_{i=1}^{n}$, the auctioneer selects the least cost in $B$, i.e., $b_{i_1,j_1} = min\{b_{i,j} | b_{i,j} \in B\}$, which is a winning bid. We set $l_{i_1,j_1} = 1$ and $W = W \cup \{d_{i_1}, t_{j_1}, b_{i_1,j_1}\}$. Then, the auctioneer removes this winning task $t_{j_1}$ and all the bids in $B$ for $t_{j_1}$, and updates $K_{i_1} = K_{i_1} - 1$. When $K_{i_1}$ is 0, the seller $d_{i_1}$ and its bids should be removed. After that, the auctioneer continues the above process until all the tasks are allocated. Finally, based on all the winning bids, the auctioneer calculates the minimum overall cost, as follows:

$$\mathbb{C}_B = \sum_{k=1}^{m} b_{i_k,j_k}, (1 \leq k \leq m). \tag{2}$$

**Step2: The Payments of Winning Bids.** The auctioneer initially restores all data to original values. For each winning bid $b_{i_k,j_k} \in W(1 \leq k \leq m)$, the auctioneer first removes this winning bid $b_{i_k,j_k}$ from $B$, and re-executes Step1 to output the minimum overall cost $\mathbb{C}_{Bn\{b_{i_k,j_k}\}}$ without the presence of $b_{i_k,j_k}$. The payment of $b_{i_k,j_k}$ is denoted by $p_{i_k,j_k}$, calculated as follows:

$$p_{i_k,j_k} = \mathbb{C}_{Bn\{b_{i_k,j_k}\}} - (\mathbb{C}_B - b_{i_k,j_k}). \tag{3}$$

### 2.3 Cryptographic Tools

**Paillier Cryptosystem.** To protect the sensitive information of buyers and sellers, we adopt Paillier homomorphic encryption scheme [16] to encrypt the sensitive data before uploading to the cloud auctioneer. A pair of key (public key $pk$ and privacy key $sk$) of this system is generated by the cloud agent. Buyers and sellers encrypt data by $E_{pk}(\cdot)$, and the agent uses $D_{sk}(\cdot)$ decrypt the ciphertext. Paillier cryptosystem has the following excellent properties: 1) *Homomorphic addition:* $D_{sk}(E_{pk}(m_1) * E_{pk}(m_2)) = m_1 + m_2$ and $D_{sk}(E_{pk}(m_1)^{m_2}) = m_1 * m_2$, where $m_1$, $m_2 \in \mathbb{Z}_n^*$, $n$ is a product of two large primes. 2) *Indistinguishability:* the same plaintext $m$ is encrypted by $pk$ multiple times, the obtained ciphertexts are different, i.e., $E_{pk}(m)_1 \neq E_{pk}(m)_2$ and $D_{sk}(m)_1 = D_{sk}(m)_2$.

**Yao's Garbled Circuits.** Yao's garbled circuits (a.k.a Yao's protocol) [17] is a general solution for secure two-party computation. The main idea is that two parties $C_1$ and $C_2$, who respectively hold their own private inputs $m_1$ and $m_2$, calculate an arbitrary function $f(m_1, m_2)$ without leaking their inputs. The main method of Yao's protocol is that $C_1$ (*circuit generator*) transforms the function $f$ into an encrypted boolean circuit (*garbled circuit*) and generates the inner circuit labels (*garbled values*) of own input $m_1$, denoted as $\widetilde{m_1}$, and then sends this garbled circuit and garbled values to the $C_2$ (*circuit evaluator*). To secretly obtain the garbled values of $C_2$'s input value $m_2$, $C_1$ and $C_2$ cooperate to execute *1-out-of-2 oblivious transfer* (OT) protocol [18]. Finally, with inputting garbled values $\widetilde{m_1}$ and $\widetilde{m_2}$, $C_1$ and $C_2$ execute the garbled circuit $f(\widetilde{m_1}, \widetilde{m_2})$, and output the result.

We briefly introduce the following garbled circuits, which have been constructed in [19]. Note that all the inputs and outputs are inner circuit labels, and the cloud servers do not learn any information from these labels.

- **XOR/AND:** The two circuits take as input an array $\{\widetilde{a_1}, \widetilde{a_2}, ..., \widetilde{a_n}\}$, where $\widetilde{a_i}$ is a $l$-bit binary, and return a $l$-bit value $\widetilde{z} = \widetilde{a_1} \oplus / \wedge \widetilde{a_2} \oplus / \wedge, ..., \oplus / \wedge \widetilde{a_n}$.
- **ADD/SUB:** The ADD/SUB circuit outputs an unsigned value of the addition/subtraction of two numbers $\widetilde{a_1}$ and $\widetilde{a_2}$, i.e., $\widetilde{z} = |\widetilde{a_1} + / - \widetilde{a_2}|$.
- **CMP:** To secretly compare the values of two numbers, we use CMP circuit input two $l$-bit binary numbers $\widetilde{a_1}$ and $\widetilde{a_2}$ to return a one-bit compared result $\widetilde{z}$. If $a \leq b$, then $z = 1$; otherwise, $z = 0$.
- **MUX:** The $MUX$ circuit is a multiplexer that has three inputs $\widetilde{a_1}$, $\widetilde{a_2}$, and an extra bit $\widetilde{\sigma}$. If $\sigma = 0$, the $MUX$ circuit outputs $a_1$; otherwise, outputs $a_2$. In this paper, we usually use this circuit to remove the invalid bids by setting $1^l$. That is, we input $\widetilde{b_{i,j}}$, $\widetilde{1^l}$, and an extra bit $\widetilde{\sigma}$, i.e., if $\sigma = 1$, we set the bid $b_{i,j} = 1^l$.

## 3 Our Protocol

### 3.1 Overview

The overview of PTATA is proposed in Algorithm 1. Specifically, the cloud agent $C_2$ generates a key pair $(pk, sk)$ of Paillier cryptosystem, and publishes $pk$.

Sellers and buyers submit the encrypted data $(E(B), E(R), E(r))$ to the cloud auctioneer $C_1$. After that, $C_1$ secretly shares these encrypted data with $C_2$ via the property of homomorphic addition, denoted as $(\langle B\rangle, \langle R\rangle, \langle r\rangle)$, in which the secret-shared value $\langle r\rangle$ is $\langle r\rangle^{C_1} = s \ mod \ 2^l$ and $\langle r\rangle^{C_2} = (r+s) \ mod \ 2^l$, $s \in \mathbb{Z}_{2^l}$ is a random number generated by $C_1$. To run the auction process in a oblivious way, $C_1$ constructs the garbled circuits of the original auction mechanism and generates the inner circuit labels of all the inputs. Finally, $C_1$ and $C_2$ cooperate to execute garbled circuits to output the actual auction results. The main algorithm of auction circuit construction will be presented later.

**Algorithm 1.** The overview of PTATA

**Input:** Sellers: the amount of resources provided $R$ and the set of bids $B$.
Buyers: the amount of resources required $r$.
**Output:** $C_1$ and $C_2$: the actual auction results.

**Phase 1: Encrypted Private Data**

1: **$C_2$:** generates a key pair $(pk, sk)$ of Paillier cryptosystem.
2: **Each Seller $d_i$:** encrypts its bids $B_i$ and available resources $R_i$ by $pk$, i.e., $E(B_i)$ and $E(R_i)$, and sends them to $C_1$.
3: **Buyers:** encrypt resources required $r$ by $pk$, and send $E(r)$ to $C_1$.

**Phase 2: Computing Garbled Circuits**

4: **$C_1$ and $C_2$:** generates the secret-shared values of all received data.
5: **$C_1$:** converts the original auction mechanism into garbled circuits, generates the inner circuit labels of its inputs, and then sends garbled circuits and garbled values to $C_2$.
6: **$C_1$ and $C_2$:** compute the garbled values of $C_2$'s inputs via OT protocol, run garbled circuits of secure winning bids determination (Sect. 3.2) and secure payments computation (Sect. 3.3), and output the actual results.

### 3.2 Secure Winning Bids Determination

After receiving the encrypted data, $C_1$ and $C_2$ execute secure winning bids determination protocol to secretly obtain all the winning bids and the overall cost. Specifically, as shown in Algorithm 2, $C_1$ first calculates the amount of tasks that each seller $d_i$ can accept with $C_2$, i.e., $K_i = min\{\lfloor \frac{R_i}{r} \rfloor, m\}(1 \leq i \leq n)$, in which $\lfloor \frac{R_i}{r} \rfloor$ can be computed via secure division computation protocol (SDC) [20] based on the property of Paillier homomorphic addition. The inputs of SDC are $E(R_i)$ and $E(r)$, and the output is the secret-shared value $\langle Rr_i\rangle$. Based on OT protocol, $C_1$ and $C_2$ obtain the garbled values $(\langle \widetilde{B}\rangle, \langle \widetilde{R}\rangle, \langle \widetilde{Rr}\rangle)$. To secretly calculate acceptable task amount $K_i$, $C_1$ and $C_2$ invoke the following TwoSMIN circuit. As shown in Fig. 2, we combine two SUB circuits and a MIN circuit to realize the desired functionality.

**TwoSMIN Circuit.** Since the secret-shared values has been transformed into garbled values, the complete values can be obtained by the SUB circuit, e.g., $\widetilde{a} = \mathrm{SUB}(\widetilde{a+s}, \widetilde{s})$, $s$ is a random number, and the MIN circuit is used to output

the minimum value between two numbers. e.g., if $a \geq b$, then MIN$(\widetilde{a}, \widetilde{b})$ outputs $\widetilde{z} = \widetilde{b}$; otherwise, outputs $\widetilde{z} = \widetilde{a}$. Note that, the MIN circuit has been proposed in [19], where $\sigma^1$ is a one-bit comparison result. Based on the MIN circuit, we construct the CMIN circuit to output this comparison result $\widetilde{\sigma^1}$, which can be used to determine the index of minimum value.

Based on TwoSMIN circuit, $C_1$ and $C_2$ can compute the acceptable task amount $\widetilde{K_i}$ = TwoSMIN$(\widetilde{Rr_i}, \widetilde{m})$ of each seller $b_i$. Next, the main process is to secretly determine the minimum bid and its index from $B$. To realize the above functionality, we build an efficient FILMIN circuit, as shown in Fig. 3. Compared with the recent work [7], the computational overhead is reduced by 50%.

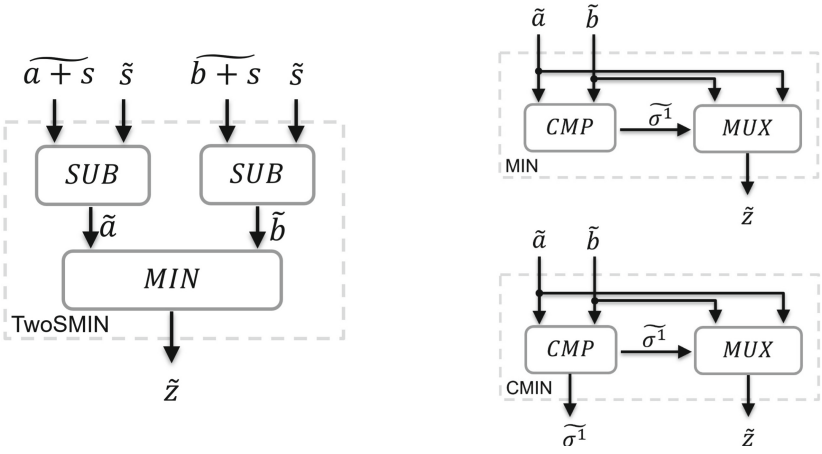



**Fig. 2.** The structure of TwoSMIN, MIN, and CMIN circuits

**FILMIN Circuit.** As shown in Fig. 3, we combine CMIN and FILTER circuits to get the minimum value and its index from an array, e.g., if an array

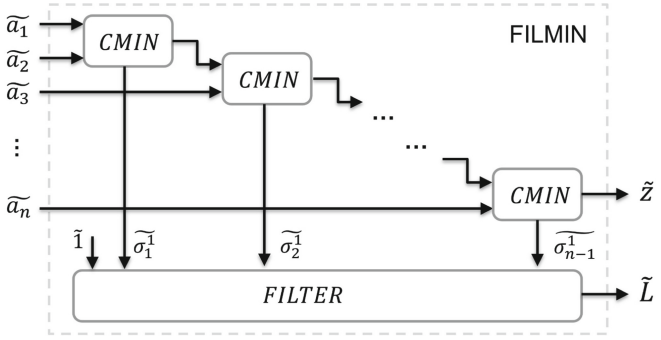



**Fig. 3.** The structure of the FILMIN circuit

$\{a_1, a_2, a_3, a_4\}$ is $\{4, 3, 3, 9\}$, then FILMIN$(\widetilde{a_1}, \widetilde{a_2}, \widetilde{a_3}, \widetilde{a_4})$ outputs the minimum value 3 and its index $\{0, 0, 1, 0\}$. The CMIN circuit that we constructed in Fig. 2 outputs the minimum value $\widetilde{z}$ and the comparison result $\widetilde{\sigma^1}$, and the FILTER circuit [21] is used to filter binary numbers, e.g., FILTER(1110) outputs $\{0,0,1,0\}$.

FILMIN circuit is executed by $C_1$ and $C_2$ to get the winning bid $\widetilde{b_{i_k,j_k}}$ and its index set $\widetilde{L_k}$. After that, $C_1$ and $C_2$ need to secretly update the acceptable task amount $K_i$ and remove the invalid bids in $B$, which are used to determine the next winning bid. The above operations are presented in detail as follows.

1 Based on index set $\widetilde{L_k}$, the indexes $(\widetilde{x_i}, \widetilde{y_j})$ of the winning seller $d_{i_k}$ and the winning task $t_{j_k}$ can be secretly calculated via XOR circuit (line 9–10), in which only $y_{j_k} = 1$ and $x_{i_k} = 1$, and others are 0.
2 After obtaining the seller $d_{i_k}$'s index set $\widetilde{x_i}$, $C_1$ and $C_2$ use SUB circuit to update the acceptable task amount $\widetilde{K_i}$, i.e., $K_i = K_i - x_i$. To secretly determine which $K_i$ is 0, they evaluate the EQ circuit to output a one-bit value $\widetilde{e_i}$. That is, if $K_i$ is 0, then $e_i = 1$; otherwise, $e_i = 0$.
3 Since all the bids of the seller $d_i$, whose $K_i$ is 0, are invalid, and all the bids for the task $t_{j_k}$ are invalid, $C_1$ and $C_2$ according to $\widetilde{e_i}$ and $\widetilde{y_j}$ can determine the invalid bids, denoted as the flag set $\widetilde{\sigma_{i,j}}$. That is, if the bid $b_{i,j}$ is invalid, then $\sigma_{i,j} = 1$; otherwise, $\sigma_{i,j} = 0$. Next, MUX circuit is used to remove all the invalid bids, denoted as $\widetilde{1^l}$, i.e., if $\sigma_{i,j} = 1$, then $b_{i,j} = 1^l$; otherwise, $b_{i,j} = b_{i,j}$.

**Algorithm 2.** Secure Winning Bids Determination

**Input:** $C_1$: Encrypted $E(B)$, $E(R)$, and $E(r)$.
$C_2$: A key pair $(pk, sk)$.
**Output:** $C_1$ and $C_2$: $\widetilde{W} = \{\widetilde{b_{i_k,j_k}}\}_{k=1}^m$, $\widetilde{L} = \{\widetilde{L_k}\}_{k=1}^m$, and the overall cost $\widetilde{\mathbb{C}_B}$.
$C_1$ **and** $C_2$**:**
1: $\langle Rr_i \rangle = SDC(E(R_i), E(r)), (\forall i \in [1, n])$.
2: Convert $(E(B), E(R))$ into secret-shared values $(\langle B \rangle, \langle R \rangle)$.
3: Generate the garbled values $(\widetilde{B}, \widetilde{R}, \widetilde{Rr})$ via OT protocol and SUB circuit.
4: $\widetilde{K_i} = \text{TwoSMIN}(\widetilde{Rr_i}, \widetilde{m}), (\forall i \in [1, n])$.
5: Initialize $\widetilde{W} = \varnothing$, $\widetilde{L} = \varnothing$, and $\widetilde{\mathbb{C}_B} = 0$.
6: **for** $k = 1$ to $m$ **do**
7: FILMIN$(\widetilde{b_{1,1}}, \widetilde{b_{1,2}}, ..., \widetilde{b_{n,m}})$, get the least value $\widetilde{b_{i_k,j_k}}$ and its index $\widetilde{L_k}$.
8: Set $\widetilde{b_{i_k,j_k}}$ as one of the winning bids.
9: Task $t_{j_k}$'s index: $\widetilde{y_j} = \text{XOR}(\widetilde{l_{1,j}}, ..., \widetilde{l_{n,j}}), (\forall j \in [1, m])$, where $\widetilde{l_{i,j}} \in \widetilde{L_k}$.
10: Seller $d_{i_k}$'s index: $\widetilde{x_i} = \text{XOR}(\widetilde{l_{i,1}}, ..., \widetilde{l_{i,m}}), (\forall i \in [1, n])$.
11: Update the acceptable task amount $\widetilde{K_i} = \text{SUB}(\widetilde{K_i}, \widetilde{x_i})$.
12: $\widetilde{e_i} = \text{EQ}(\widetilde{K_i}, 0)$, if $K_i$ is 0 that the seller $d_i$ should be remove, set $e_i = 1$.
13: Find the invalid bid indexes in $\widetilde{B}$ via $\widetilde{y_j}$ and $\widetilde{e_i}$, denoted as $\widetilde{\sigma_{i,j}}$.
14: Remove all the invalid bids $\widetilde{b_{i,j}} = \text{MUX}((\widetilde{b_{i,j}}, \widetilde{1^l}), \widetilde{\sigma_{i,j}})$.
15: $\widetilde{L} = \widetilde{L} \cup \widetilde{L_k}$ and $\widetilde{W} = \widetilde{W} \cup \{\widetilde{b_{i_k,j_k}}\}$.
16: **end**
17: Calculate the overall cost $\widetilde{\mathbb{C}_B} = \text{ADD}(\widetilde{b_{i_1,j_1}}, \widetilde{b_{i_2,j_2}}, ..., \widetilde{b_{i_m,j_m}})$.
18: **return** $\widetilde{W} = \{\widetilde{b_{i_k,j_k}}\}_{k=1}^m$, $\widetilde{L} = \{\widetilde{L_k}\}_{k=1}^m$, and $\widetilde{\mathbb{C}_B}$.

After that, $C_1$ and $C_2$ put this winning bid value $\widetilde{b_{i_k,j_k}}$ and its index set $\widetilde{L_k}$ into the set $\widetilde{W}$ and $\widetilde{L}$, respectively. They continue the above process until all the tasks are allocated. Finally, they use a ADD circuit to calculate the overall cost $\widetilde{\mathbb{C}_B} = \text{ADD}(\widetilde{b_{i_1,j_1}}, \widetilde{b_{i_2,j_2}}, ..., \widetilde{b_{i_m,j_m}})$, and then output $\widetilde{W}$, $\widetilde{L}$, and $\widetilde{\mathbb{C}_B}$.

### 3.3 Secure Payments Computation

Based on the winning bids determined in the above subsection, $C_1$ and $C_2$ need to secretly calculate the payment for each winning bid. Specifically, as shown in Algorithm 3, they initially restore $(\widetilde{B}, \widetilde{R}, \widetilde{K})$ to the initial garbled values. For each winning bid $\widetilde{b_{i_k,j_k}}$ in $\widetilde{W}$, they first use MUX circuit to remove $\widetilde{b_{i_k,j_k}}$ from $\widetilde{B}$ and then re-execute Algorithm 2 to get the another overall cost $\widetilde{\mathbb{C}}_{Bn\{b_{i_k,j_k}\}}$. Based on Eq. (2), two SUB circuits are used to obtain the payment $\widetilde{p_{i_k,j_k}}$. After that, $C_1$ and $C_2$ continue the above process until all the payments are calculated. Finally, they reveal all the winning sellers and buyers $(d_{i_k}, t_{j_k})$ and the payments $p_k$. The payments for other bids that do not win are 0.

**Algorithm 3.** Secure Payments Computation

**Input:** $C_1$ and $C_2$: $\widetilde{W} = \{\widetilde{b_{i_k,j_k}}\}_{k=1}^{m}$, $\widetilde{L} = \{\widetilde{L_k}\}_{k=1}^{m}$, and $\widetilde{\mathbb{C}_B}$.
**Output:** $C_1$ and $C_2$: The actual auction results $(d_{i_k}, t_{j_k}, p_{i_k,j_k})$.

$C_1$ **and** $C_2$**:**
1: Restore $(\widetilde{B}, \widetilde{R}, \widetilde{K})$ to the initial values.
2: **while** $\forall \widetilde{b_{i_k,j_k}} \in \widetilde{W}$ **do**
3: Remove the winning bid $\widetilde{b_{i,j}} = \text{MUX}((\widetilde{b_{i,j}}, 1^l), \widetilde{l_{i,j}})$, where $\widetilde{l_{i,j}} \in \widetilde{L_k}$.
4: Execute Alg. 2(line 5-18) to output $\widetilde{\mathbb{C}}_{Bn\{b_{i_k,j_k}\}}$ without $\widetilde{b_{i_k,j_k}}$.
5: \\ Calculate $p_{i_k,j_k} = \mathbb{C}_{Bn\{b_{i_k,j_k}\}} - (\mathbb{C}_B - b_{i_k,j_k})$.
6: $\widetilde{p_{i_k,j_k}} = \text{SUB}(\widetilde{\mathbb{C}}_{Bn\{b_{i_k,j_k}\}}, \text{SUB}(\widetilde{\mathbb{C}_B}, \widetilde{b_{i_k,j_k}}))$.
7: **end**
8: Reveal the results of winners $(d_{i_k}, t_{j_k})$ and the payments $p_{i_k,j_k}$.
9: The payments for other bids that do not win are 0.

### 3.4 Security and Efficiency Analysis

**Security Analysis.** Based on composition theory [15], we prove the cryptography security of PTATA under the semi-honest model [14].

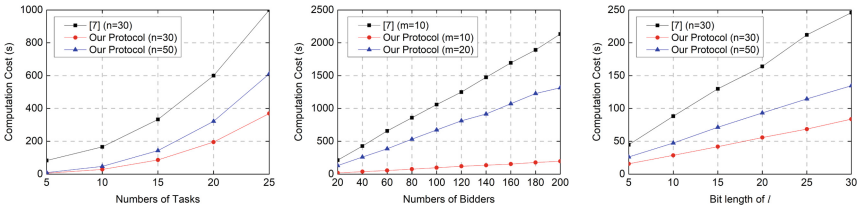
**Theorem 1.** *As long as Paillier cryptosystem and various circuits are secure under the semi-honest model, PTATA is secure under the semi-honest model.*

***Proof.*** On the one hand, Since $C_2$ is responsible for generating the key pair $(pk, sk)$ of Paillier cryptosystem, $C_1$ cannot decrypt the encrypted data. Before obtaining the secret-shared values, $C_1$ uses homomorphic addition to randomize

these encrypted data, which sent to $C_2$. Since Paillier cryptosystem has been proved to be semantically secure, $C_1$ and $C_2$ cannot learn anything from these encrypted data and secret-shared values. On the other hand, various circuits including XOR, EQ, MIN, SUB, CMP, MUX and TwoSMIN, and FILMIN, are both applied in Yao's garbled circuits, and all intermediate values are inner circuit labels. Since Yao's garbled circuits have been proved to be secure under the semi-honest model [22], PTATA is secure under the semi-honest model.

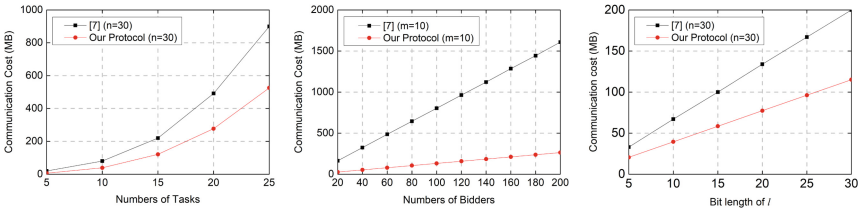
**Efficiency Analysis.** The main cost is garbled circuits for execution. Fortunately, the XOR gate has almost no overhead with "free XOR" technique [23], and the efficiency of our system depends on the amount of non-XOR gates. In Algorithm 2, the main process is to determine the winning bids and remove the invalid bids, and the efficiency of this process is $O(nm^2l)$, where $l$ is the bit length of each bid. In Algorithm 3, the main process is to calculate the overall cost without the winning bid, and the efficiency of this process is $O(nm^3l)$.

## 4 Experiments




(a) Numbers of Tasks $m$ (b) Numbers of Bidders $n$ (c) Bit length of $l$

**Fig. 4.** Computation cost induced by PTATA.




(a) Numbers of Tasks $m$ (b) Numbers of Bidders $n$ (c) Bit length of $l$

**Fig. 5.** Communication cost induced by PTATA.

We implement our scheme in FastGC [21], which is a Java-based framework. The cloud servers $C_1$ and $C_2$ are both simulated on an Intel i5-11600H CPU, 3.90GHz,

and 16GB RAM computer. The security level of inner circuit labels for garbled circuits is 80-bit and the security modulus of Paillier cryptosystem is 1024-bit. In our experiments, we mainly evaluate the computation and communication costs for different bidder numbers $n$, task numbers $m$, and bit lengths $l$. The default settings of $n$, $m$, and $l$ are 30, 10, and 10, respectively.

As shown in Fig. 4 and Fig. 5, we compare the computation and communication costs of our protocol with the recent work [7]. We can see that the increasing trends of the costs for different $n$, $m$, and $l$ are consistent with our analysis $O(nm^3l)$. Note that, our protocol is more efficient than this recent work. Specifically, in Figs. 4(a) and 5(a), when $n = 30$ and $m$ changes from 5 to 25, the costs of our protocol increase from (5.7s, 6.1 MB) to (369.5s, 526.1 MB), but the costs of the recent work increase from about (80s, 20MB) to (990s, 900MB). In Figs. 4(b) and 5(b), when $m = 10$ and $n = 200$, our protocol (195.8 s, 263.6 MB) only costs about 9.1% computation overhead and 16.3% communication overhead of the recent work, respectively. In summary, the experiment results show that the costs of our protocol are acceptable in practical applications.

## 5 Related Work

Recently, there are various works to deal with privacy-preserving cloud auction, Specifically, Chen *et al.* [5] first design a privacy-preserving cloud auction for Virtual Machines (VMs) allocation based on a data-oblivious way, which protects the privacy of bidders. However, this scheme does not consider available resources privacy. Then, Cheng *et al.* [6] propose an efficient and secure double cloud auction scheme, which can protect the privacy of all users. Nevertheless, the challenges of this solution (such as secure compare-and-swap and secure sorting) are different from our problem. Besides, the recent work [7] proposes a secure auction scheme for heterogeneous task assignment, but this scheme does not consider the number of task required privacy.

Another related research topic is privacy-preserving auction for spectrum allocation. Chen *et al.* [8,9] propose two privacy-preserving double auction schemes for homogenous spectrum allocation based on a series of secure arithmetic public-key operations. To improve the system efficiency, the work [10] designs a secure spectrum auction scheme via public-key encryption system, but this scheme does not consider bidders' location information privacy. To protect location information, Wang *et al.* [11,12] propose a series of privacy-preserving and truthful auction schemes for double spectrum auction. Unfortunately, these schemes disclose the access patterns privacy. Recently, Cheng *et al.* [13] propose a lightweight framework, which ensures high efficiency while providing a strong security guarantee. However, this solution involves large amounts of precomputed multiplication triplets between two cloud servers, which is unrealistic.

## 6 Conclusion

In this paper, we have proposed PTATA, a privacy-preserving and truthful auction scheme for task assignment in outsourced cloud environments. Moreover,

we have proved that PTATA protocol is secure under the semi-honest model and have analyzed the system efficiency. Based on extensive experiments, our solution is acceptable in real-life applications.

**Acknowledgement.** This work was partially supported by Natural Science Foundation of China (Grant No. 61602400) and Jiangsu Provincial Department of Education (Grant NO. 16KJB520043).

## References

1. Wang, X., Chen, X. Wu, W.: Towards truthful auction mechanisms for task assignment in mobile device clouds. In: Proceedings of Conference on Computer Communications (INFOCOM), pp. 1–9. IEEE, Atlanta (2017)
2. Jin, A.-L., Song, W., Zhuang, W.: Auction-based resource allocation for sharing cloudlets in mobile cloud computing. Trans. Emerg. Top. Comput. **6**(1), 1–12 (2015)
3. Wang, A.-L., et al.: Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing. Trans. Serv. Comput. **9**(6), 1–14 (2015)
4. Shi, J., Yang, Z., Zhu, J.: An auction-based rescue task allocation approach for heterogeneous multi-robot system. Multimed. Tools App. **79**, 14529–14538 (2018). https://doi.org/10.1007/s11042-018-7080-4
5. Chen, Z., et al.: On privacy-preserving cloud auction. In: Symposium on Reliable Distributed Systems, pp. 279–288. IEEE, Budapest (2016)
6. Cheng, K., et al.: Towards efficient privacy-preserving auction mechanism for two-sided cloud markets. In: ICC, pp. 1–6. IEEE (2019)
7. Jiang, X., Pei, X., Tian, D., Li, L.: Privacy-preserving auction for heterogeneous task assignment in mobile device clouds. In: Liu, Z., Wu, F., Das, S.K. (eds.) WASA 2021. LNCS, vol. 12939, pp. 345–358. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86137-7_38
8. Chen, Z., et al.: PS-TRUST: provably secure solution for truthful double spectrum auctions. In: INFOCOM, pp. 1249–1257. IEEE, Toronto (2014)
9. Chen, Z., et al.: Secure, efficient and practical double spectrum auction. In: IWQoS, pp. 1–6. IEEE, Vilanova (2017)
10. Wang, J., et al.: A secure spectrum auction scheme without the trusted party based on the smart contract. Dig. Commun. Netw. **7**(2), 223–234 (2021)
11. Wang, Q., et al.: Privacy-preserving and truthful double auction for heterogeneous spectrum. TON **27**(2), 848–861 (2019)
12. Wang, Q., Huang, J., Chen, Y., et al.: Prost: Privacy-preserving and truthful online double auction for spectrum allocation. Trans. Inf. Forensics Secur. **14**(2), 374–386 (2019)
13. Cheng, K., et al.: A lightweight auction framework for spectrum allocation with strong security guarantees. In: INFOCOM, pp. 1708–1717. IEEE, Toronto (2020)
14. Goldreich, O.: Foundations of Cryptography, vol. 2, Basic Applications. Cambridge University Press, Cambridge (2004)
15. Liu, A., et al.: Efficient secure similarity computation on encrypted trajectory data. In: ICDE, pp. 66–77. IEEE (2015)
16. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16

17. Yao, A.C.: How to generate and exchange secrets. In: FOCS, pp. 162–167 (1986)
18. Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending oblivious transfers efficiently. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 145–161. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_9
19. Kolesnikov, V., Sadeghi, A.-R., Schneider, T.: Improved garbled circuit building blocks and applications to auctions and computing minima. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 1–20. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10433-6_1
20. Cui, N., et al.: SVKNN: efficient secure and verifiable k-nearest neighbor query on the cloud platform. In: ICDE, pp. 253–264. IEEE (2020)
21. Huang, Y., et al.: Faster secure two-party computation using garbled circuits. In: USENIX Security, San Francisco (2011)
22. Lindell, Y., Pinkas, B.: A proof of security of Yao's protocol for two-party computation. J. Cryptol. **22**(2), 161–188 (2009)
23. Kolesnikov, V., Schneider, T.: Improved garbled circuit: free XOR gates and applications. In: Aceto, L., et al. (eds.) ICALP 2008. LNCS, vol. 5126, pp. 486–498. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70583-3_40

# An SM2-based Traceable Ring Signature Scheme for Smart Grid Privacy Protection

Da Teng[1,2,3], Yanqing Yao[1,2,3,4(✉)], Yingdong Wang[3], Lei Zhou[3], and Chao Huang[1,2,3]

[1] State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
[2] State Key Laboratory of Cryptology, Beijing 100878, China
[3] Key Laboratory of Aerospace Network Security, Ministry of Industry and Information Technology, School of Cyber Science and Technology, Beihang University, Beijing 100191, China
[4] Beijing Key Laboratory of Network Technology, Beihang University, Beijing 100191, China
{yaoyq,zb2039109}@buaa.edu.cn

**Abstract.** A smart grid can dynamically adjust the amount of electricity supply by smart meters' personalized needs, reducing energy waste and protecting the environment. However, because the uploaded data could reveal users' sensitive information, and internal adversaries could poison the power statistics, privacy preservation and security supervision in smart grid systems need to be concerned. To solve these, we propose a traceable ring signature scheme based on SM2 with strong security and anonymity, in addition to utilizing this scheme to build a four-layer smart grid model, separating the duty of statistics and regulations. Specifically, the scheme integrates the advantages of a key-insulated linkable ring signature (LRS) for Monero and an SM2-based ring signature: a key derivation mechanism to make the key more secure and a simple SM2-based ring structure. A trapdoor has been introduced in the "key image" of the signature, which is often used in LRS for linkability, but in our signature, it's used for traceability. This allows authorized participants to open signatures and reveal the identity of the real signer when exceptions occur. Besides the security and privacy analyses, we also implement the proposed scheme and give some experiments to evaluate the time and space performance. The results show that our new scheme with space of kilobyte level size and time of linear or constant cost can be effectively

This work is supported by the National Key Research and Development Program of China (No. 2021YFB3100400), the National Natural Science Foundation of China (grant no. 62072023), Beijing Municipal Natural Science Foundation (grant no. 4202035), the Open Project Fund of the State Key Laboratory of Cryptology (grant no. MMKFKT202120), the Exploratory Optional Project Fund of the State Key Laboratory of Software Development Environment (grant no. SKLSDE-2020ZX-23), and the Fundamental Research Funds of Beihang University(grant nos. YWF-20-BJ-J-1040 and YWF-21-BJ-J-1041).


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 296–313, 2022.
https://doi.org/10.1007/978-3-031-19208-1_25

adapted to the functional requirement of our smart grid model. In addition, the signature can be ported to wireless mobile devices for privacy protection and security management.

**Keywords:** Smart grid · Privacy protection · Traceable ring signature · Chinese commercial cryptography

## 1 Introduction

A smart grid is a large-scale infrastructure combining information technology, power transmission, automatic control technology and other technologies. It is considered the next-generation power supply network, which transmits electricity reliably and efficiently from generators to the consumer side [21]. The composition of a smart grid generally includes a control center, substations, smart meters and the transmission network between them. Through this network, electricity is supplied on demand at fluctuating prices, reducing energy waste and protecting the environment [8]. Consumers can turn on household appliances or electrical equipment when electricity is cheap and turn off some non-essential appliances during peak periods to reduce demand [7].

The most significant difference between a smart grid and a traditional grid is that it can realize two-way information services. Users can know the real-time power supply status, price fluctuations and other information from smart meters and make personalized electricity plans according to the information. Meanwhile, the control center of the smart grid can collect electricity consumption data through its centralized network and adjust power supply and price dynamically based on the statistical results. Smart grids have been recognized by many countries in the world as a measure that can effectively reduce energy waste and the global greenhouse effect. The European Union, the United States and other governments actively promote the smart grid development model [1]. They have made meaningful progress in organizational structure, incentive policies, standard systems, and critical technology research and development.

However, there are still some problems in the practical application of smart grids. First of all, there is the privacy issue. Users' electricity plans may reflect sensitive information such as living habits, family population and capacity of factories, which may lead to some users' reluctance to use smart meters and affect the popularity of smart gird [15]. The second is the exception handling issue. As an extensive infrastructure network, the smart grid must be able to take measures to detect and handle exceptions while avoiding privacy issues caused by power abuse. For example, when receiving an exception, the centralized nodes of a smart grid need to locate specific data and trace the uploader's identity, but malicious internal employees of the power company could trace the private information of legitimate users through abuse of traceability.

We can protect users' privacy from two aspects of smart grids to solve these problems. One is data encryption, which encrypts power data and uploads it as ciphertext. Data encryption techniques, such as homomorphic encryption,

can be used to calculate statistics while protecting the data content of a single user. The other is Anonymous identity. Since data upload is periodic in smart grids, the change rule of the same user's power data in a period may reveal habits. Authentication can be obfuscated to disrupt the relevance between each round of upload data while satisfying system functional requirements such as traceability.

### 1.1 Contributions

This paper proposes a new approach to the two main challenges of smart grid: privacy protection and traceability. The contributions of this paper include:

- We propose a four-layer smart grid model based on the classical three-layer model [8].
- To implement the model, we propose a traceable and privacy-preserving signature scheme for smart gird, called stealth address traceable ring signature (SA-TRS). We utilize stealth address linkable ring signature(SA-LRS) [12] to anonymize users' identity, where a new ephemeral address is hidden in a set of ring members. We also make the signatures traceable by introducing a pair of trapdoor keys in the key image.
- Under the ECDLP assumption, we construct the scheme based on a Chinese commercial cryptographic algorithm SM2 [3] and give proof of security and privacy.
- We perform Functional and performance experiments. The results shows that our SA-TRS could satisfy the system's functional requirements and protect security and privacy at a small cost of time and space.

The rest of the paper is organized as follows. Related work is reviewed in Sect. 2, and the definition of the formal model and requirements are given in Sect. 3. Some preliminaries are introduced in Sect. 4. Our SM2-based traceable SA-LRS construction and its proof of correctness are proposed in Sect. 5. The proof of security and privacy are given in Sect. 6, followed by the performance evaluation of our scheme in Sect. 7. Finally, Sect. 8 lists extensible applications and Sect. 9 concludes the paper.

## 2 Related Work

There has been much research on the privacy protection of smart grids, mainly focusing on encrypted data aggregation. Nonetheless, it is also common to protect identity/address through anonymity, and research has been proposed that replacing the real identity with a fake one in smart grid [17]. However, existing researches usually transfer identity in a straightforward way, like concatenating some random nonce following the real identity [8] which is not the real anonymity or introduces complex computation that affects the load balancing and computational efficiency. For example, generate credentials by non-interactive zero-knowledge(NIZK) [20]. Besides, existing research fails to keep the duties of data collection and traceable supervision separate [19].

Similarly, Monero, a cryptocurrency focuses on private transactions [6], adopts stealth address proposed in CryptoNote [2] to protect the real address(i.e.public keys involved in transaction). Liu et al. [13] pointed out that the insulation between master identity and derived identity as well as derived identities themselves be required; otherwise, there will be a risk of being broken. Moreover, the ring signature was first proposed by Rivest, Shamir and Tauman in 2001 [16]. It is a type of signature that provides very strong anonymity, where the signer can generate signatures on behalf of a group of potential signers [14]. Researchers have applied it to the trusted authentication and privacy protection of smart grids [18]. Liu et al. [12] proposed a lattice-based linkable ring signature scheme that supports stealth address(SA-LRS) which satisfies key insulation and proves its security and privacy under the random oracle model.

## 3 Formal Model and Requirements

### 3.1 Formal Model

To achieve stronger privacy protection under traceability, we draw inspiration from the separation-of-duties (SoD) for access control, which separates the duty of aggregation and traceability into different departments. Specifically, we modify the traditional three-layer model of smart grids to a four-layer model. Figure 1 shows the architecture of our smart grid model, which we improved from Chim et al.'s [8] architecture, which is a centralized network composed of three functional facilities, including smart meters, substations and a control center, in addition to the power and communication lines between them. Power lines transmit electricity from the power station to the local substations, which process and distribute it to terminals (corresponding to the red lines in Fig. 1). Communication lines transmit meter data to local substations and send the aggregated data packages to the control center (corresponding to the blue line in Fig. 1). We introduce a supervision module for monitoring and tracking so that any abnormal statistical result can be traced back to the single uploader (corresponding to the green line in Fig. 1). The four facilities involved in our model have defined as follows:

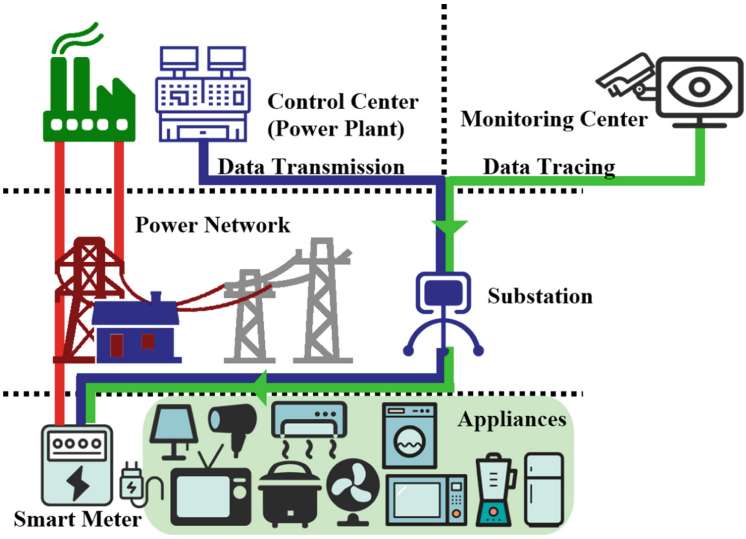



**Fig. 1.** Our Four-layer Architecture of Smart Gird

**Smart Meter.** An intelligent device with finite computing power helps users make electricity plans and uploads signed data to the substation. For simplicity, this paper uses smart meters as the same component as users, with no distinction.

**Substation.** A department that collects data uploaded by the electricity meters under its jurisdiction packages it up and uploads it to the control center after verification. As the intermediary between smart meters and the control center, one substation is connected with many meters, and multiple substations correspond to one control center.

**Control Center.** A department that collects real-time total electricity plans uploaded by substations and adjusts the generation. It is also responsible for some additional statistics and administration.

**Monitoring Center.** That is the innovative part of our model, which is the main difference between our model and others. A department that detects exceptional data and traces it back to the smart meter that sent it. The monitoring center knows the trapdoor secret key of the signing algorithm so that the actual signer can be easily found. Otherwise, that would be very difficult.

Note that, as an independent module, we regard the monitoring center as a trusted third party, which in practice should be a law enforcement agency independent of the power company because it holds the secret key (i.e. the trapdoor) to open the data envelope. That is designed to prevent privacy disclosure caused by collusion.

We propose a particular signature scheme to construct this model to achieve responsibility traceability and identity anonymity. The signature scheme needs to satisfy:

- Unforgeability: Any PPT adversary cannot generate a valid signature without knowing the secret key.

- Anonymity: The advantage of any PPT adversary in distinguishing the real signer is negligible.
- Traceability: An authorized participant can open the envelope without interaction.

### 3.2 Formal Signature

We improve the SA-LRS proposed by Liu et al. [12] into a traceable ring signature, the formal definition of our scheme is almost the same as [12], except for the function of $Trace(\cdot)$. The formal definition of our traceable ring signature with stealth address(SA-TRS) includes the following algorithms:

- $\boldsymbol{Setup(1^{\lambda}) \rightarrow PP}$. This is a probabilistic algorithm. On inputting the security parameter $\lambda$, the algorithm outputs the system public parameters $PP$.
- $\boldsymbol{MasterKeyGen(PP) \rightarrow (msk_i, mpk_i)}$. This is a probabilistic algorithm. On inputting the system parameters $PP$, the algorithm outputs a pair of master keys (master secret key, master public key). $i$ is the index of the signer, which is kept secret.
- $\boldsymbol{DerivedPublicKeyGen(mpk_i) \rightarrow dpk_{i_j}}$. This is a probabilistic algorithm. On inputting a master public key $mpk_i$, the algorithm outputs a derived public key $dpk_{i_j}$. $j$ is the index of signer $i$'s derived keys, which is also kept secret.
- $\boldsymbol{DerivedPublicKeyCheck(dpk'_{i_j}, (mpk_i, msk_i)) \rightarrow 0/1}$. This is a deterministic algorithm. On inputting a derived key $dpk'_{i_j}$ and a pair of master keys $(mpk_i, msk_i)$, the algorithm checks whether it was a valid derived key generated from this master key pair.
- $\boldsymbol{Sign(m, R, dpk_{i_j}, (mpk_i, msk_i)) \rightarrow \sigma}$. This is a probabilistic algorithm. On inputting message $m$, a ring of valid derived public keys $R = \{dpk_{i_1}, ..., dpk_{i_t}\}$ ($t$ is the size of ring), the signer's derived public key $dpk_{i_j} \in R$ and corresponding master key pair $(mpk_i, msk_i)$, the algorithm outputs a signature $\sigma$.
- $\boldsymbol{Verify(m', R, \sigma) \rightarrow 0/1}$. This is a deterministic algorithm. On inputting a message $m'$, a ring of valid derived public keys $R$ and a signature $\sigma$, the algorithm verifies whether it's a valid signature for message $m'$ generated from the ring.
- $\boldsymbol{Trace(\sigma, sk_{trapdoor}) \rightarrow mpk_i}$. This is a deterministic algorithm. On inputting a signature $\sigma$ and the secret trap key, the algorithm traces the real signer and outputs its master public key $mpk_i$ if it's valid. Otherwise, output $\perp$.

In a linkable ring signature, a key image $I$ is usually used to link the signatures generated by the same address (public key), but in our scheme, the key image is used for a different purpose. In the data tracing phase of our model, the monitoring center knows the trapdoor's secret key. It can decrypt the key image by calling the $Trace$ function to obtain the signer's identity information. For anyone who does not know the trapdoor, it is difficult to relate different signatures of the same signer, and for anyone who knows the trapdoor, any signature is traceable. In the preparation phase, each signer generates its master

key pair, which represents the real identity of the signer. In the data uploading phase, our model adopts SA-TRS to sign the message, and each signature uses a newly generated derived public key. Through the key derivation mechanism: One is that the adversary can not ensure who the real signer is since he/she can only get a set of pseudo addresses. Second is that, different from LRS; the adversary can not link different signatures of the same signer together so that the extra information would not be disclosed from the changing pattern of power data.

### 3.3 Security and Privacy Requirements

In this part, we define the security and privacy model of SA-TRS. Liu et al. have defined the strong unforgeability and anonymity for SA-LRS [12], which are also applicable to SA-TRS. We review the definition of strong unforgeability and anonymity as follows: **Strong Unforgeability** [12] For any probabilistic polynomial time (PPT) adversary $\mathcal{A}$, the advantage of $A$ to win the following game is negligible:

- **Setup phase**
  Run $Setup(1^{\lambda}) \rightarrow PP$ and send $PP$ to $A$. Run $MasterKeyGen(PP) \rightarrow (msk_i, mpk_i)$ and send $mpk_i$ to $A$. $A$ initializes a set $L_{Dpk}$, which is used to store derived keys that have been queried and validated.
- **Query phase**
  (1) OAdd:
  On receiving a pair of $(dpk'_{i_j}, mpk_i)$ from $A$, run $DerivedPublicKey\text{-}Check(dpk'_{i_j}, (mpk_i, msk_i)) \rightarrow 0/1$. If the output is 1, add the key pair to $L_{Dpk}$.
  (2) OSign: On receiving $(M, R, dpk_{i_j} \in R \cap L_{Dpk})$, $OSign$ runs $Sign(\cdot)$ and returns $\sigma$ to $A$.
- **Output phase** $A$ outputs a tuple of message, ring and signature: $(M^*, R^*, \sigma^*)$.

$A$ wins this game when:

(1) $Verify(M^*, R^*, \sigma^*) = 1$.
(2) $R^* \in L_{Dpk}$.
(3) The tuple $(M^*, R^*, \sigma^*)$ has not been queried.

**Anonymity** [12]

- **Setup phase.** Same as above.
- **Query phase.** Same as above.
- **Challenge phase.** On receiving tuple $(M^*, R^*, (dpk_0, dpk_1 \in R^* \cap L_{Dpk}))$ from $A$. Randomly choose a bit $b \in \{0, 1\}$. Run $Sign(M^*, R^*, dpk_b)$ and return $\sigma$. Note that $dpk_0$ and $dpk_1$ have never been signed before, 0 and 1 are just the indexes of some derived public keys.
- **Query phase.** Same as above, except that $dpk_0$ and $dpk_1$ can not be queried.
- **Guess phase.** $A$ guesses a bit $b'$.

$A$ succeed if the advantage in guessing $b$ correctly is not negligible. The advantage can be represented as $|Pr[b' = b] - \frac{1}{2}|$.

## 4 Preliminaries

Before we propose our signature scheme, we first review some underlying tools and schemes. The security of SA-TRS is based on the hardness of the elliptic curve discrete logarithm problem, whose definition is reviewed in this part. Our ring signature scheme is based on SM2 and its variant signature schemes. SM2 is a public key cryptographic algorithm based on elliptic curves, which has become the Chinese commercial cryptographic standard and is mainly used to take the place of the RSA algorithm in the Chinese commercial cryptographic system. Another basis for SA-TRS is the key isolation proposed in [12], which enhances the key security of the scheme. Moreover, We review the forking lemma commonly used in signature security proofs.

### 4.1 EC-DLP

Let $E(F_q)$ be an elliptic curve defined over a finite field $F_q$. $E : y^2 = x^3 + ax + b$, $a, b \in F_q$. Point $G \in E(F_q)$ of order $n$, point $P \in$ Cyclic group generated based on basis $G$. The Elliptic-curve Discrete-logarithm problem consists of finding an integer $k \in [0, n-1]$ such that

$$P = kG \tag{1}$$

The security of SA-TRS depends on the hardness of EC-DLP.

**Algorithm 1.** SM2 digital signature algorithm

**procedure** SETUP($\lambda$):
  define $H_v$: a cipher hashing function with a message digest length of $v$ bits;
  define $Z_A$ : an identity hash value;
  $G$: the generator of elliptic curve;
  **return** param;
**end procedure**
**procedure** KEY GENERATION:
  randomly choose $s \in [1, n-2]$, compute $P = sG$;
  **return** $(s, P)$;
**end procedure**
**procedure** SIGN($M$, $s$):
  set $\bar{M} = Z_A \| M$;
  compute $e = H_v(\bar{M})$;
  randomly choose $k \in [1, n-1]$;
  compute point $(x_i, y_i) = kG$;
  compute $r = e + x_i (mod\ n)$, back to line 15 if $r = 0$ or $r + k = n$;
  compute $z = (1+s)^{-1} \cdot (k - r \cdot s)(mod\ n)$, back to line 15 if $z = 0$;
  **return** signature $\sigma = (r, z)$;
**end procedure**
**procedure** VERIFY($P$, $M'$,$(r', z')$):
  Check if $r' \in [1, n-1]$ holds, if not, return 0;
  Check if $z' \in [1, n-1]$ holds, if not, return 0;
  Set $\bar{M}' = Z_A \| M'$;
  Compute $e' = H_v(\bar{M}')$;
  Compute $t = r' + z'(mod\ n)$. If $t = 0$, return 0;
  Compute point $(x'_1, y'_1) = z'G + tP$;
  Compute $R = e' + x'_1 (mod\ n)$;
  **if** $R = r'$ **then**
    **return** 1;
  **else**
    **return** 0;
  **end if**
**end procedure**

### 4.2 Chinese Commercial Cryptographic SM2 [3,5]

SM2 contains a digital signature scheme, as Algorithm 1 shows.

SM2 also contains a public key encryption scheme [3], which is treated as a black box in our scheme. Its formal definition is as follows:

**User's key-pair** private key $d_B \in [1, n-2]$ is chosen uniformly, public key $P_B = d_B G$, $G$ is the basis of elliptic curve.

**Encrypt.** The algorithm $Enc_{P_B}(M, r) \rightarrow C$ encrypts the plaintext $M$ with the public key $P_B$, $r$ is a random parameter.

**Decrypt.** The algorithm $Dec_{d_B}(C) \rightarrow M$ decrypts the ciphertext $C$ with the private key $d_B$ and gets the plaintext.

### 4.3 SM2-based Ring Signature

Han et al. constructed a ring by $c$ instead of $z$ in step 6 of algorithm $SM2.Sign(\cdot)$ in [10]. The algorithm defines $c$ as Algorithm 2 shows.

**Algorithm 2.** Han's SM2-based ring signature algorithm (the definition part of ring)

1: set $z_j = (1 + s_j)^{-1} \cdot (k_j - c_j \cdot s_j)(mod\ n)$;
2: **for** $i = j+1, \ldots, t, 1, \ldots, j-1$ **do**
3: set $z_i = (r_i + c_i)P_i + r_i G$;
4: set $c_{i+1} = H(P_i, M, z_i)$;
5: **end for**
6: **return** $\sigma = (c_1, r_1, \ldots, r_t)$;

According to the generic construction of standard ring signature types generalized by Abe et al., this is a ring signature of type-T, which constructs a C-ring by hash function during the signing. It would be reconstructed during verification. Besides, Fan et al. [9] proposed another linkable scheme. The ring structure of our scheme refers to these two schemes and makes it traceable.

### 4.4 Liu's Key Isolation Scheme with KEM [12]

Liu et al. noticed the risks of key disclosure from stealth addresses in Monero, because of the un-insulation between derived keys, between derived keys and master keys. They proposed a lattice-based linkable ring signature scheme that combines key derivation and insulation by utilizing the randomness introduced by the Key Encapsulation Mechanism(KEM). The formal definition of KEM algorithms:

- $SetUp(1^{\mu}) \rightarrow PP_{kem}$.
- $KeyGen(PP_{kem}) \rightarrow (pk_{kem}, sk_{kem})$.

- $Encaps(pk_{kem}) \rightarrow (c, \kappa)$.
- $Decaps(c, pk_{kem}, sk_{kem}) \rightarrow \kappa$.

Our SA-TRS uses this isolation approach to protect the master keys. For security purposes, the KEM algorithm of SM9 [4] (another Chinese commercial cryptographic algorithm) is employed, which is treated as a black box in our scheme.

### 4.5 Forking Lemma for Ring Signature

Javier and German [11] generalized the forking lemma to the ring signatures' scenario. The lemma is generally used in security proof of ring signatures.

**Generic Ring Signature Schemes**. Consider a security parameter $k$, a hash function $hash(\cdot)$ which outputs k-bit long elements, and a ring $A_1, ..., A_n$ of $n$ members. A ring signature tuple $\Sigma_{ring} : (m, R_1, ..., R_n, h_1, ..., h_n, \sigma)$, where $m$ is the message to be signed, $R_1, ..., R_n$ are the parameters that provide randomness with $R_i \neq R_j$ for all $i \neq j$, $h_i$ is the value of $hash(m, R_i)$ for $1 \leq i \leq n$ and $\sigma$ is the signature that fully determined by above parameters.

The following lemma is the version considering chosen-message attacks defined in a generic ring signature.

**Forking Lemma**. For a probabilistic polynomial time Turing machine adversary $\mathcal{A}$. Define $Q$ and $W$ as the number of queries that $\mathcal{A}$ can ask to the random oracle and to some real signers, respectively. Within time $T$, if $\mathcal{A}$ can produce a valid signature $(m, R_1, ..., R_n, h_1, ..., h_n, \sigma)$ with some non-negligible probability, then two valid signatures $(m, R_1, ..., R_n, h_1, ..., h_n, \sigma)$ and $(m, R_1, ..., R_n, h'_1, ..., h'_n, \sigma)$ can be produced, such that $h_j \neq h'_j$ for some $j \in 1, .., n$ and $h_i = h'_i$ for all $i = 1, .., n$ with $i \neq j$.

## 5 SM2-based SA-TRS Scheme

This section presents the construction of our improved SA-TRS scheme and proves its correctness.

### 5.1 Construction

We propose an SM2-based traceable ring signature scheme with stealth addresses, satisfying user privacy and traceability in our smart grid model. We combines the key isolation derivation [12] with the ring structure based on SM2 [9,10]. For standardization of the tools being called in the scheme, different Chinese commercial cryptographic algorithms are used as auxiliary algorithms, including the SM9 key encapsulation mechanism, the SM2 public key encryption algorithm and the SM3 cryptographic hash algorithm. All the SM2 and other Chinese commercial cryptographic algorithms involved have been indexed in the ISO/IEC standards list.

- $\boldsymbol{Setup(1^{\lambda}) \rightarrow PP}$ The algorithm sets the system public parameters $PP = \{q, a, b, G, p\}$, including the size $q$ of finite field $F_q$, the parameters $a$ and $b$ of the elliptic curve $E(F_q)$, the basis $G$ of $E(F_q)$ and its order $p$. The algorithm also includes:
  (1) Let $Kem_{SM9}$ denote the SM9 key-encapsulation mechanism scheme under the hardness of EC-DLP. Run $Kem_{SM9}.SetUp(1^{\mu}) \rightarrow PP_{kem}$ to initialize system parameters.
  (2) Let $PKE_{SM2}$ be the SM2 public key encryption algorithm. Run $PKE_{SM2}.KeyGen(PP_{pke}) \rightarrow (pk_{trapdoor}, sk_{trapdoor})$.
  (3) There are several hash functions contains in this scheme, which be viewed as random oracles. They are $H_1(\cdot) : \mathcal{K}_{kem} \rightarrow Z_p$, a cryptographic hash $H_v(\cdot)$ with a message digest length of v bits and $H_{SM3}(\cdot)$.
- $\boldsymbol{MasterKeyGen(PP) \rightarrow (msk_i, mpk_i)}$ The algorithm does:
  (1) Select a random number $s_i \in Z_p$. Compute $P_i = s_i G$.
  (2) Run $Kem_{SM9}.KeyGen(PP_{kem}) \rightarrow (pk_{kem_i}, sk_{kem_i})$ and get a pair of KEM keys.
  (3) Output $mpk_i = (P_i, pk_{kem_i})$, $msk_i = (s_i, sk_{kem_i})$.
- $\boldsymbol{DerivedPublicKeyGen(mpk_i) \rightarrow dpk_{i_j}}$ The algorithm does:
  (1) Run $Kem_{SM9}.Encaps(pk_{kem_i}) \rightarrow (c_{i_j}, \kappa_{i_j})$ to encapsulate a KEM public key and get the ciphertext $c_{i_j}$ and the secret key $\kappa_{i_j}$.
  (2) Set $s^*_{i_j} \leftarrow H_1(\kappa_{i_j})$. Compute $P^*_{i_j} = s^*_{i_j} G$ and $\hat{P}_{i_j} = P^*_{i_j} + P_i$.
  (3) Output $dpk_{i_j} = (c_{i_j}, \hat{P}_{i_j})$.
- $\boldsymbol{DerivedPublicKeyCheck(dpk_{\cdot_j}, (mpk_i, msk_i)) \rightarrow 0/1}$ The algorithm does:
  (1) If $c_{\cdot_j} \notin \mathcal{C}_{kem}$ or $\hat{P}_{\cdot_j}$ doesn't satisfies the elliptic curve equation, stop and output 0.
  (2) Run $Kem_{SM9}.Decaps(c_{\cdot_j}, pk_{kem_i}, sk_{kem_i}) \rightarrow \kappa'_{i_j}$ to decapsulate and get secret key $\kappa'_{i_j}$.
  (3) Recover $s^{*'}_{i_j} \leftarrow H_1(\kappa'_{i_j})$ and $P^{*'}_{i_j} = s^{*'}_{i_j} G$.
  (4) If $\hat{P}_{i_j} = P_i + P^{*'}_{i_j}$ holds, output 1; otherwise, output 0.
- $\boldsymbol{Sign(m, R, dpk_{i_j}, (mpk_i, msk_i)) \rightarrow \sigma}$ Let $i$ be the index of the signer, $R = \{dpk_{1\cdot}, \ldots, dpk_{t\cdot}\}$ be a ring of valid derived public keys. The algorithm includes two phases: I: Generate the key image
  (1) Run $Kem_{SM9}.Decaps(c_{i_j}, pk_{kem_i}, sk_{kem_i}) \rightarrow \kappa_{i_j}$ to decapsulate and get the secret key $\kappa_{i_j}$.
  (2) Run $H_1(\kappa_{i_j}) \rightarrow s^*_{i_j}$, $\hat{s}_{i_j} = s^*_{i_j} + s_i$.
  (3) Run $PKE_{SM2}.Enc_{pk_{trapdoor}}(mpk_i \| dpk_{i_j} \| s^*_{i_j}) \rightarrow I$. $I$ is the ciphertext of $PKE_{SM2}$.

  Phase II: Generate the signature
  (1) Choose a random $k_{i_j} \in Z_p$. Set $C_{i+1} = H_{SM3}(m, R, k_{i_j} G, I)$
  (2) Set $\bar{m} = z_A \| m$ and $e = H_v(\bar{m})$.
  (3) For $l = i+1, \ldots, t, 1, \ldots, i-1$
    (a) Randomly choose $k_{l\cdot} \in Z_p$.
    Compute $k_{l\cdot} G$ the corresponding elliptic curve point $(x_l, y_l)$.
    (b) Set $r_l = e + x_l (mod\ p)$ and convert it to an integer.

(c) Set $z_l = (r_l + C_l \cdot I)\hat{P}_{l\cdot} + r_l G$.
(d) Set $C_{l+1} = Hash_{SM3}(m, R, z_l, I)$. Mark $C_{t+1}$ as $C_1$.
(4) Set $r_i = (1 + \hat{s}_{i_j})^{-1}(k_{i_j} - C_i \cdot \hat{s}_{i_j} \cdot I)(mod\ p)$.
(5) Output $\sigma = (C_1, r_1, \ldots, r_t, I)$.

– $\boldsymbol{Verify(m', R, \sigma) \rightarrow 0/1}$ The algorithm does:
(1) If $C_1' \notin Z_q^*$ holds, stop and output 0.
(2) For $l = 1, \ldots, t$
If $r_i' \notin Z_q^*$, stop and output 0.
(3) For $l = 1, \ldots, t$
(a) Deconstruct $R$ and get the l-th derived public key $P_{l\cdot}$.
(b) Set $z_l' = (r_l' + C_l' \cdot I') \cdot \hat{P}_{l\cdot} + r_l' G$.
(c) Set $C_{l+1}' = H_{SM3}(m', R, z_l', I)$.
(4) If $C_{t+1}' = C_1$, output 1; otherwise, output 0.

– $\boldsymbol{Trace(\sigma, sk_{trapdoor}) \rightarrow mpk_i}$ The algorithm does:
(1) Run $PKE_{SM2}.Dec_{sk_{trapdoor}}(I) \rightarrow Text$. $Text = (mpk_i \| dpk_{i_j} \| s_{i_j}^*)$. Split $Text$ into $mpk_i = (P_i, pk_{kem_i})$, $dpk_{i_j} = (c_{i_j}, \hat{P}_{i_j})$ and $s_{i_j}^*$.
(2) Compute $P_{i_j}^* = s_{i_j}^* G$.
(3) If $P_{i_j}^* + P_i = \hat{P}_{i_j}$, output $mpk_i$. Otherwise, output $\perp$.

### 5.2 Proof of Correctness

**Correctness**. Start from $C_1$, the algorithm compute $z_l'$ and $C_{l+1}'$ sequentially:

$$\begin{cases} z_1' = (r_1 + C_1 \cdot I')\hat{P}_{1\cdot} + r_1 G \\ C_2' = Hash_{SM3}(m', R, z_1', I') \end{cases} \tag{2}$$

$$\begin{cases} z_2' = (r_2 + C_2' \cdot I')\hat{P}_{2\cdot} + r_2 G \\ C_3' = Hash_{SM3}(m', R, z_2', I') \end{cases} \tag{3}$$

...

$$\begin{aligned} z_i' &= (r_i + C_i' \cdot I')\hat{P}_{i_j} + r_i G \\ &= \frac{k_{i_j} - C_i \cdot \hat{s}_{i_j} \cdot I + C_i' \cdot I' + C_i' \cdot \hat{s}_{i_j} \cdot I'}{1 + \hat{s}_{i_j}} \hat{P}_{i_j} + \frac{k_{i_j} - C_i \cdot \hat{s}_{i_j} \cdot I}{1 + \hat{s}_{i_j}} G \\ &= \frac{(k_{i_j} + C_i' \cdot I') \cdot \hat{s}_{i_j} + k_{i_j} - C_i \cdot \hat{s}_{i_j} \cdot I}{1 + \hat{s}_{i_j}} G \\ &= k_{i_j} G \end{aligned} \tag{4}$$

$$C_{i+1}' = Hash_{SM3}(m', R, k_{i_j} G, I') \tag{5}$$

...

$$\begin{cases} z'_t = (r_t + C'_t \cdot I')\hat{P}_{t.} + r_t G \\ C'_{t+1} = Hash_{SM3}(m', R, z'_t, I') = C'_1 \end{cases} \tag{6}$$

Notice that these equations imply $C_j \cdot I = C'_j \cdot I'$. Only if every $z_i, C_{i+1}$ is recovered the ring correctly would be approved. For any PPT adversary without any knowledge of the secret trap key, each ring member is indistinguishable.

## 6 Proof of Privacy and Security

### 6.1 Proof of Anonymity

According to the game of anonymity in Sect. 3.3, the analysis is as follows:

**Theorem 1.** *The SA-TRS scheme is anonymous in the random oracle model.*

***Proof.*** During challenge phase, the adversary $A$ query and get a valid signature $\sigma' = (C'_1, r'_1, ..., r'_t, I')$. Assuming that the advantage of $A$ guessing $b$ correct is not negligible, that is, $A$ can distinguish the signing key of $\sigma'$ with non-negligible probability.

(1) $I'$ is ciphertext encrypted with the trapdoor, thus $A$ cannot distinguish encrypted information through $I'$.
(2) $C'_i (i \in 1, ..., t)$ is a group of hash values that are indistinguishable due to the unidirectionality of the hash.
(3) $r'_i (i \in 1, ..., t)$ is a group of integers in $Z_n$ consisting of $t-1$ random values and another identically distributed value. Different $r'_i$ are statistically indistinguishable.

In summary, the assumption does not hold. So our scheme satisfies anonymity.

### 6.2 Proof of Strong Unforgeability Based on Forking Lemma

**Theorem 2.** *The SA-TRS scheme is strongly unforgeable in the random oracle model.*

***Proof.*** Since the ring structure is similar, we can use the proof method based on the forking lemma in [9] to prove the strong unforgeability given in Sect. 3.3.

Assuming that the advantage of $A$ to succeed is not negligible, which means a valid tuple $(M^*, R^*, \sigma^*)$ can be output in PPT time. Using ring signature forking lemma, we can get another tuple $(M^*, R^*, \sigma^{*'})$ with equal random parameters $r^*_1, ..., r_{i-1}, k^*_{i_j}, r_{i+1}, ...r^*_t$ and some $j$ has different hash parameter $c^*_j \neq c^{*'}_j$.

Since both signatures are valid, there are:

$$k^*_{i_j} G = (r^*_i + C^*_i \cdot I^*)\hat{P}_{i_j} + r^*_i G \tag{7}$$

$$k_{i_j}^* G = (r_i^{*'} + C_i^{*'} \cdot I^{*'})\hat{t}_{i_j} + r_i^{*'} G \tag{8}$$

From (2)-(3), we can compute:

$$\hat{P}_{i_j} = \frac{r_i^{*'} - r_i^*}{r_i^* - r_i^{*'} + C_i^* \cdot I^* - C_i^{*'} \cdot I^{*'}} G \tag{9}$$

We know $I^*, I^{*'}, r_i^*, r_i^{*'}$, besides $C_i^*$ and $C_i^{*'}$ are computable. Obviously, the difficulty of ECDLP is broken by $\hat{s}_{i_j}$:

$$\hat{s}_{i_j} = \frac{r_i^{*'} - r_i^*}{r_i^* - r_i^{*'} + C_i^* I^* - C_i^{*'} I^{*'}} \tag{10}$$

So the assumption does not hold. The advantage of $A$ to break the game is negligible.

## 7 Evaluation

### 7.1 Functionality Comparison

In this part, the functionality comparison of our scheme with some other similar papers. We give the illustration on mainly four differences: key insulation, identity anonymity, traceability and privacy computing, as shown in Table 1.

**Table 1.** Functionality comparison.

| Papers | Key Insulation | Identity anonymity | Traceability | Privacy computing |
|---|---|---|---|---|
| PASS11 [8] | | √ | √ | |
| Yang17 [21] | | | √ | √ |
| Wu20 [19] | | | √ | √ |
| SM2-RS [10] | | √ | | |
| SA-LRS [12] | √ | √ | | |
| Tang21 [18] | | √ | √ | √ |
| **Ours** | √ | √ | √ | √ |

In reference to that four properties: Key insulation means that it is irrelative between the meter's master key and the temporary one. This property represents stronger key security protection. Identity anonymity demands that anyone with no auxiliary information cannot recognise the actual owner of the data packet. Traceability denotes that there exists a trusted party is competent to trace down the collected data to the owner's identity. Privacy computing in the smart grid allows the control center to get statistics but not specific data that is widely regarded as implying private information. A popular solution is homomorphic

encryption, and all the schemes shown in Table 1 that satisfy this property, including ours, adopted this idea.

Among all the papers mentioned, our scheme is the only one that supports all four conditions.

### 7.2 Performance Evaluation

This part gives the performance evaluation of our proposed scheme for both time and space.

We implement the proposed scheme by building a java project on Eclipse Java EE IDE (Oxygen Release 4.7.0) based on Java 1.8.0. These experiments were run on a laptop ASUS FL8000UF8550 installing Windows 10 Home China, Intel(R) Core(TM) i7-8550U CPU @1.80 GHz 1.99 GHz with Memory 8 GB.

Without loss of generality, we evaluate our scheme with a standard elliptic curve family NIST over prime fields, including $NIST\_P\_192$, $NIST\_P\_224$, $NIST\_P\_256$, $NIST\_P\_384$ and $NIST\_P\_521$. While to control variables, we just use $NIST\_P\_192$ in the time experiment, and we fix the ring size with 10 in the other experiment. Each experiment is repeated 40 times to achieve stable results and takes the average result.

About the reason for using NIST family curves as test parameters: Although our signature is constructed based on Chinese cryptographic standards, we still want to evaluate them under some common parameter systems.

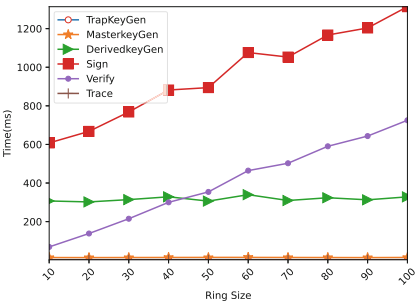



**Fig. 2.** Time cost for each step

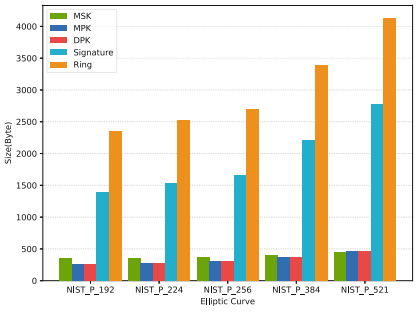



**Fig. 3.** Size comparison for each step

**Time Performance Test.** According to Fig. 2, the line graph shows the time cost (in milliseconds) for different steps as the ring size changes between 10 and 100. It is clear that the consuming time saw a significant increase at a near-linear trend in the two steps of signing and verifying by the ring size growing, while the time costs of the rest steps rise slightly or follow just stable trends. The time consumption for signing is the leading cost over the whole period,

reaching about 1.3 s with ring size 100, and the cost of the verifying process shows an almost parallel tendency with it. In contrast, the time cost of trapdoor generation, master key generation and tracing steps are negligible.

**Space Performance Test.** As far as Fig. 3, the bar chart compares the size of keys and the signature of our scheme under different elliptic curves being adopted. Note that we exclude the trapdoor key because we use the ready-made public-key cryptosystem (SM2), whose keyspace is determined by its public parameters, so this is an irrelevant factor.

It is widely known that the marked number of the NIST elliptic curve represents the bit length of the large prime number used. All the five classes of bars in the chart saw a rise as the marked number increased; three of them are consistently under 500 bytes and have only a slight rise, namely the size of the master secret key, the master public key and the derived public key. Hence our scheme can control the scale of the key at a microscopic level with a stable trend, while the results of the rest two bars are more significant and show a more pronounced increase. Along with the different elliptic curves from tagged 192 to 521, the scales of signature and ring increase approximately from 1.4 to 2.8kb and from 2.4 to above 4kb, respectively.

As for the above results, they are consistent with the theoretical structure of our ring signature scheme. On the one hand, the size of the ring can only affect the time complexity at a polynomial level of the signing and validation processes, but for the rest processes, it is irrelevant. On the other hand, the space cost of our scheme is on the order of kilobytes when using the standard curves from the NIST family.

## 8 Applications

Moreover, due to the anonymity and traceability of SA-TRS, it can be ported to wireless mobile devices and network communication services to balance privacy protection and security management. Let's give some applications: First of all, the signature scheme can be used for location privacy protection in mobile communication environments such as the internet of vehicles(IoV). Besides, it can help to offer multi-party mobile communication services such as anonymous wireless meetings and anonymous multi-party micropayments. Additionally, our application in smart grid can combine with the smart home, for privacy-friendly wireless smart power applications.

## 9 Conclusion

This paper proposes a four-layers smart grid model, which separates the duty of statistics and regulations. We propose a new SM2-based traceable ring signature scheme with strong security and anonymity by integrating and improving existing schemes. Experiments have been performed to test this signature scheme's time and space performance. With those advantages, the signature can help

achieve traceable responsibility and anonymous identity in our proposed grid model. By joint with other techniques such as homomorphic encryption and differential privacy to protect data content, dual protection can be provided for identity and sensitive data. Future research can focus on the following issues: Optimize the structure of the signature and further improve the space-time efficiency. Moreover, due to the anonymity and traceability of SA-TRS, it can be used in wireless mobile devices and network communication services such as the IoV.

## References


1. American recovery and reinvestment act of 2009.https://www.congress.gov/bill/111th-congress/house-bill/1
2. Cryptonote v 2.0. https://bytecoin.org/old/whitepaper.pdf
3. GM/T 0003–2012 public key cryptographic algorithm SM2 based on elliptic curves. http://www.oscca.gov.cn/sca/xxgk/2010-12/17/1002386/files/b791a9f908bb4803875ab6aeeb7b4e03.pdf
4. GM/T0044-2016 identity-based cryptographic algorithms SM9. http://www.oscca.gov.cn/sca/xxgk/2016-03/28/content_1002407.shtml
5. RFC 8998 ShangMI (SM) cipher suites for TLS 1.3. https://www.rfc-editor.org/info/rfc8998
6. Zero to Monero: First edition. https://www.getmonero.org/library/Zero-to-Monero-1-0-0.pdf
7. Chim, T.W., Yiu, S.M., Li, V.O., Hui, L.C., Zhong, J.: PRGA: privacy-preserving recording and gateway-assisted authentication of power usage information for smart grid. IEEE Trans. Depend. Secur. Comput. **12**(1), 85–97 (2015)
8. Chim, T., Yiu, S., Hui, L.C., Li, V.O.: Pass: privacy-preserving authentication scheme for smart grid network. In: 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 196–201 (2011)
9. Fan, Q., He, D., Luo, M., Huang, X., Li, D.: Ring signature schemes based on SM2 digital signature algorithm. J. Cryptol. Res. **8**(4), 710–723 (2021)
10. Han, B., Li, Z.: Research and design of ring signature scheme based on SM2 cryptographic algorithm. Commun. Technol. **54**(7), 1–12 (2021)
11. Herranz, J., Sáez, G.: Forking lemmas for ring signature schemes. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 266–279. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-24582-7_20
12. Liu, Z., Nguyen, K., Yang, G., Wang, H., Wong, D.S.: A lattice-based linkable ring signature supporting stealth addresses. In: Sako, K., Schneider, S., Ryan, P.Y.A. (eds.) ESORICS 2019. LNCS, vol. 11735, pp. 726–746. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29959-0_35
13. Liu, Z., Yang, G., Wong, D.S., Nguyen, K., Wang, H.: Key-insulated and privacy-preserving signature scheme with publicly derived public key. In: IEEE European Symposium on Security and Privacy (EuroS&P), pp. 215–230 (2019)
14. Lu, X., Au, M.H., Zhang, Z.: Raptor: a practical lattice-based (linkable) ring signature. In: Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds.) ACNS 2019. LNCS, vol. 11464, pp. 110–130. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21568-2_6

15. Ming, Y., Li, Y., Zhao, Y., Yang, P., Yao, Y.: Efficient privacy-preserving data aggregation scheme with fault tolerance in smart grid. Secur. Commun. Netw. **2022**, 1–18 (2022)
16. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_32
17. Srinivas, J., Das, A.K., Li, X., Khan, M.K., Jo, M.: Designing anonymous signature-based authenticated key exchange scheme for internet of things-enabled smart grid systems. IEEE Trans. Ind. Inform. **17**(7), 4425–4436 (2021)
18. Tang, F., Pang, J., Cheng, K., Gong, Q.: Multiauthority traceable ring signature scheme for smart grid based on blockchain. Wirel. Commun. Mob. Comput. **2021**(1), 1–9 (2021)
19. Wu, F., Li, X., Xu, L., Kumari, S.: A privacy preserving scheme with identity traceable property for smart grid. Comput. Commun. **157**, 38–44 (2020)
20. Xia, X., Ji, S.: An efficient anonymous authentication scheme for privacy-preserving in smart grid. In: 2021 IEEE Conference on Dependable and Secure Computing (DSC), pp. 1–2 (2021)
21. Yang, Q., Hong, J., Xue, K., Chen, W., Zhang, X., Yue, H.: A privacy-preserving and real-time traceable power request scheme for smart grid. In: 2017 IEEE International Conference on Communications (ICC), pp. 1–6 (2017)

# Collusion-Tolerant Data Aggregation Method for Smart Grid

Liyuan Cao[1], Yingwen Chen[1], Kaiyu Cai[1], Dongsheng Wang[2(✉)], Yuchuan Luo[1], and Guangtao Xue[3]

[1] National University of Defense Technology, Changsha, China
[2] Academy of Military Sciences, Beijing, China
wdsh_2015@163.com
[3] Shanghai Jiao Tong University, Shanghai, China

**Abstract.** In smart grid, smart meters record real-time customers' electricity consumption and transmit it to the control center. Leakage of customers' real-time power usage data will impose severe risk on consumers' privacy. To Protect users' privacy, researchers observe that what the control center really needs for providing services is not raw data but aggregate values of customers' real-time usage data. On this basis, many privacy-preserving data aggregation schemes are proposed for smart grid. However, these schemes pay little attention to the problem of collusion. In these schemes, edge nodes store the ciphertexts of customers' raw usage data, and the control center has the decryption key. If edge nodes collude with the control center, customers' usage data will still be leaked. To tackle this problem, a collusion-tolerant and privacy-preserving data aggregation scheme for smart grid is proposed in this paper. By leveraging homomorphic encryption and threshold scheme, the proposed scheme supports typical aggregate operations in smart grid, while keeping the privacy of customers and defending the collusions from edge nodes and the control center in practice. Security analysis and performance evaluation demonstrates that the proposed scheme protects the privacy of customers' usage data effectively even under collusions in practice.

**Keywords:** Smart grid · Paillier homomorphic encryption · Shamir threshold scheme

## 1 Introduction

Smart grid is built on the integrated communication network, which integrates all electricity transmission networks connected to grid users to effectively provide sustainable, economic and safe electricity. With its advanced sensing and measurement, smart grid has attracted the attention of both industry and academia. By 2027, Western Europe's energy sector will invest more than $100 billion in smart grid construction [18].


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 314–325, 2022.
https://doi.org/10.1007/978-3-031-19208-1_26

Smart meters collect customers' real-time power usage data. The control center analyses the aggregated results of these data and adjusts customers' electricity consumption mode and guides the customers' electricity usage by dynamically formulating the electricity price. For example, when the electricity price is raised during the peak period, customers will use less electricity consumption, so that the control center can reduce the pressure on electricity transmission. From the customers' point of view, real-time electricity consumption information can reflect customers' living habits and other activities. For example, persistent low levels of electricity use indicate that no one is home. When electricity consumption data becomes low at night, the family may be sleeping. There is no doubt that real-time electricity usage data should be private to the customer.

Customers want to protect their privacy data from threats despite the relevant commitments made by smart grid companies. Researchers notice what the control center of smart grid really needs for providing services is not the original power usage data but the aggregated results. There are many data aggregation methods to protect customer privacy. Homomorphic encryption is a common method to solve data aggregation problems. Smart meters encrypt electricity consumption data and send them to cloud nodes or edge nodes. Cloud nodes or edge nodes perform the corresponding data aggregation calculation, and then deliver the aggregation results to the control center for decryption. By this means, the control center can only get the aggregate value of customers' electricity consumption data, not the specific value. Edge nodes has the ciphertext of customers' electricity consumption data, but it does not have the key to decrypt, nor can it know the specific electricity consumption data of customers. These schemes can protect privacy and satisfy the control center's need for aggregated data at the same time.

However, another problem has appeared. If the control center and edge nodes conspired, many schemes will become vulnerable. The control center has decryption key and edge nodes have ciphertexts of customers' original consumption data. If they conspire, edge nodes send original encrypted data without aggregation to the control center, which can decrypt and obtain customers' real-time power usage data. Customers' privacy is still at risk of exposure. Fortunately, We notice that threshold scheme can solve the security problem caused by collusion effectively. The threshold scheme divides data into many fragments, and only by collecting a amount of fragments can the original data be recovered. Combining homomorphic encryption and threshold scheme, we designed a method to solve the privacy problem caused by the collusion of the control center and edge nodes.

Other chapters are arranged as follows. We review the related work in Sect. 2. Problem statement is described in Sect. 3. We introduce preliminaries that will be used in our scheme in Sect. 4. We present the scheme that tolerate collusion in smart grid in Sect. 5. The safety and performance are analyzed respectively in Sect. 6 and 7. Finally, we make a conclusion in Sect. 8.

## 2 Related Work

Much attention has been paid to the study of data aggregation in IoT. We review some data aggregation schemes in the following.

**Schemes Based on Cryptography.** Gai et al. [3] designed a data aggregation method using random response to satisfy locally differentiated privacy. Gope et al. [5] designed a scheme using lightweight authentication encryption to adapt to resource constrained devices. Jianbing et al. [12] designed an identity authentication mechanism using hash function to protect smart grid from pollution attack. Gong et al. [4] focus on cryptographic primitives such as partially blind signature and design a scheme to verify customers' identity using signature measurement data. PDMA [18] can resist malicious data mining attacks in smart grid through a knowledge signature mechanism. Luo et al. [10] propose a decentralized microgrid data aggregation scheme by leveraging PBFT consensus. EBDA [9] combines edge computing with blockchain to improve security. [1,2,7] propose data aggregation schemes for wireless sensing networks with complex topologies or big sensory data.

**Schemes Based on Edge Computing.** There is also a lot of work focusing on edge or cloud computing [13,21]. Zhao et al. [22] set a edge node between customers and the control center based on edge computing. It feeds the results to the control center. PPFA [11] distributes noise generation among parties through Gaussian mechanism to reduce privacy leakage and be efficient. Saleem et al. design a fault-tolerant scheme FESDA [15], which can work when nearly half of smart meters fail. FPDA [20] fixes the vulnerability that the control center in FESDA [15] can obtain the reading of smart meter. Liu et al. [8] present a scheme that supports aggregation communication and function query with a double trapdoor decryption cryptosystem. Wang et al. [17] used pairing based signature to preserves customers' privacy and ensures data integrity. APPA [6] uses pseudonym certificates and pays attention to the anonymity and authenticity of devices in data aggregation. A distributed fog computing coordinator [19] is introduced, which collects the information of edge computing nodes regularly. The coordinator enables fog nodes to collaborate on more complex tasks.

Frameworks combined with edge computing offer significant advantages in protecting customers' private data. They separate the customer's privacy data from the data aggregation process. The above work has solved the problem of privacy protection in data aggregation at many levels, but the actual situation may be more complex. Many such frameworks do not notice the collusion between control center and edge nodes, which is very unfavorable to privacy protection. In order to make up for the neglect of conspiracy attack in the current work, we design a scheme that can resist collusion attack in smart grid.

# 3 Problem Statement

## 3.1 System Model

Trusted authority(TA), control center(CC), edge nodes(EN) and smart meters(SM) constitute the main system of smart grid. The model is shown in Fig. 1. The functions of each part are as follows:

**Trusted Authority (TA):** The trusted authority registers the joined nodes and distributes the key (public key and private key). It will go offline after smart grid starts working.

**Control Center (CC):** The control center connects to all edge nodes and receives data from them. The control center analyzes the aggregated customers' real-time power usage data customers and provide dynamic pricing and other services.

**Edge Node (EN):** Edge nodes are located between the control center and customers. Edge nodes receive data from smart meters, and perform preliminary aggregation. They are also used to forward the request between control center and customers. An edge node can be a wireless access device, which is required to be able to calculate and store data. We suppose that there are $n$ edge nodes in our smart grid, expressed as $EN = \{EN_1, EN_2, EN_3, \cdots, EN_n\}$. The data edge nodes deliver to the control center are expressed as $N = \{N_1, N_2, N_3, \cdots, N_n\}$.

**Smart Meters (SM):** Each customer has a smart meter at home. They collect and process customers' real-time power usage data, and finally hand over the data to the control center. We suppose that there are $m$ smart meters in our smart grid, expressed as $SM = \{SM_1, SM_2, SM_3, \cdots, SM_m\}$, and the readings shown in smart meters are expressed as $M = \{M_1, M_2, M_3, \cdots, M_m\}$.

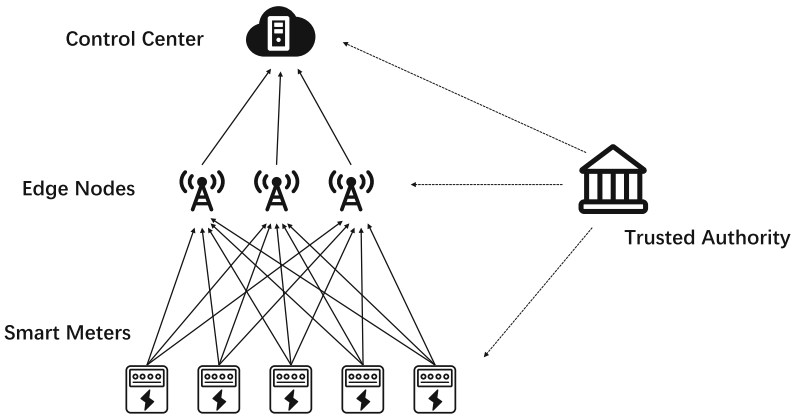



**Fig. 1.** System model for data aggregation in smart grid.

## 3.2 Security Assumption

Customers want to ensure that their real-time power usage data is not known by other parts of the smart grid. We assume that TA is completely reliable. It accurately registers nodes, issues keys, and goes offline after the smart grid is built,

and it will not be attacked. CC is honest but curious. It will issue instructions to smart meters as agreed to collect aggregated data, but it also wants to get the real-time power usage data of customers. It may take the following means: Calculate the data it can get to restore the electricity consumption data of a specific customer; Or collude with one or several edge nodes to get the electricity consumption data before aggregation operation of the edge nodes, and decrypt a specific customer's electricity consumption data through the key. Edge nodes may not be able to send messages when attacked. Edge nodes are also curious, which may collude with the control center to get the decrypted key and try to decrypt the customers' real-time electricity consumption information.

## 4 Preliminaries

### 4.1 Paillier Homomorphic Encryption

With homomorphic encryption, users can perform calculations on the encrypted data and then decrypt it. These calculation results are retained in encrypted form. When decrypted, the results are the same as the output generated when operations are performed on unencrypted data. In the following, we briefly review the Paillier homomorphic encryption [14] that we adopt in this paper.

1. Key generation
   Select two large prime numbers $p$ and $q$ satisfying
   $$gcd(pq, (p-1)(q-1)) = 1 \tag{1}$$
   Calculate $n$ and $\lambda$
   $$n = pq \tag{2}$$
   $$\lambda = lcm(p-1, q-1) \tag{3}$$
   Define function $L(x)$
   $$L(x) = \frac{(x-1)}{n} \tag{4}$$
   Select an integer $g$ less than $n^2$ randomly, and there is
   $$\mu = [L(g^\lambda \bmod n^2)]^{-1} \bmod n \tag{5}$$
   We choose the public key $pk = (n, g)$ and the private key $sk = (\lambda, \mu)$.
2. Encryption
   Select an integer $r = Z^*_{n^2}$, compute ciphertext $c$:
   $$c = g^m r^n \bmod n^2 \tag{6}$$
3. Decryption
   $$m = L(c^\lambda \bmod n^2) \cdot \mu \bmod n \tag{7}$$

The homomorphic properties of this cryptosystem can be described as:

$$\begin{aligned} Enc(m_1 + m_2) &= g^{m_1+m_2}(r_1 r_2^n) \bmod n^2 \\ &= Enc(m_1) \cdot Enc(m_2) \end{aligned} \tag{8}$$

$$Enc(a \cdot m_1) = g^{am_1} r_1^{an} \bmod n^2 = Enc(m_1)^a \tag{9}$$

### 4.2 Shamir Threshold Scheme

Threshold scheme try to divide a secret data $D$ into $n$ parts $D_1, D_2, \cdots, D_n$ to achieve the purposes that $D$ can be accurately calculated when knowing any $k$ or more $D_j$, but $D$ cannot be calculated when knowing any $k-1$ or fewer $D_j$. Shamir threshold scheme [16] is an effective threshold scheme.

1. Secret fragment generation
   Randomly pick a $k-1$ degree polynomial:

$$q(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1} \bmod p \tag{10}$$

   In this polynomial, $a_0 = D$, and evaluate $D_1 = q(1), \cdots, D_j = q(j), \cdots, D_n = q(n)$. Give $n$ group $(j, D_j)$ to $n$ people, open $p$ and destroy the polynomial $q(x)$.
2. Secret recovery
   Select any $k$ different secret pieces and the polynomial will be recovered by Lagrange interpolation:

$$q(x) = \sum_{i=1}^{k} D_i \prod_{1 \leqslant j \leqslant k, j \neq i} \frac{x - x_j}{x_j - x_i} \bmod p \tag{11}$$

   The value of $q(0)$ is the secret data $D$.

## 5 The Proposed Scheme

We propose a scheme that both meets the data aggregation needs of the control center, and protects the privacy information of customers' real-time power consumption. Specifically, what we can do is aggregation protocol for sum, max/min, and electricity bill.

### 5.1 Aggregation Protocol for Sum

The control center can aggregate different customers' electricity consumption data. In other words, the control center can calculate $\sum_{s=1}^{m} M_s$.

1. Each smart meter divides the electricity consumption data into $n$ parts depending on shamir threshold scheme. Specifically, according to Sect. 3, the $i$-th smart meter is expressed as $SM_i$ and the reading of $SM_i$ is expressed as $M_i$. $SM_i$ divides $M_i$ into $n$ parts expressed as $\{M_{i,1}, M_{i,2}, M_{i,3}, \cdots, M_{i,n}\}$ according to the $k-1$ degree polynomial $q_i(x)$ it picks.

$$q_i(x) = M_i + a_{i,1} x + a_{i,2} x^2 + \cdots + a_{i,k-1} x^{k-1} \tag{12}$$

$$M_{i,j} = M_i + a_{i,1} \cdot j + a_{i,2} \cdot j^2 + \cdots + a_{i,k-1} \cdot j^{k-1} \tag{13}$$

2. Customers use the public key $pk$ to encrypt the divided data. Specifically, $\{M_{i,1}, M_{i,2}, M_{i,3}, \cdots, M_{i,n}\}$ are encrypted to $\{Enc(M_{i,1}),\ Enc(M_{i,2}),\ Enc(M_{i,3}),\ \cdots,\ Enc(M_{i,n})\}$.
3. Next, customers need to send the encrypted divided data to the edge nodes. It should be noted here that the $j$-th divided data of each customer shall be sent to the $j$-th edge node. For example, the edge node $EN_j$ will receive encrypted split data from all customers, which should be $\{Enc(M_{1,j}),\ Enc(M_{2,j}),\ Enc(M_{3,j}),\ \cdots,\ Enc(M_{m,j})\}$.
4. Each edge node receives $m$ pieces of split data, which comes from different customers. And then each edge node calculates the product of these split data. Specifically, the calculation result $N_j$ of the $j$-th edge node $EN_j$ should be

$$N_j = \prod_{s=1}^{m} Enc(M_{s,j}) \tag{14}$$

   According to the homomorphism of paillier cryptosystem, the multiplication of these encrypted split data is equivalent to their summation and then encryption. All edge nodes send their calculation results to the control center.
5. The control center randomly selects $k$ edge nodes from $n$ edge nodes. We express these $k$ edge node as $\{EN_{x_1}, EN_{x_2}, EN_{x_3}, \cdots, EN_{x_k}\}$. The control center calculates the interpolation basis function through the number of the selected $k$ edge nodes. Then, the power operation is carried out according to the obtained result and the value passed by the edge node. Finally, the sum of the customers' electricity consumption data can be obtained by decrypting with the private key $sk$. **Algorithm 1** describes these actions of the control center.

$$CC = Dec(\prod_{i=1}^{k} N_{x_i}^{\prod_{1 \leqslant j \leqslant k, j \neq i} \frac{-x_j}{x_i - x_j}}) = \sum_{s=1}^{m} M_s \tag{15}$$

**Algorithm 1.** Sum and Bill Aggregation Algorithm of CC

**Input:** Calculation results of all edge nodes $N_1$, $N_2$, $N_3$, $\cdots$, $N_n$; private key $sk$
**Output:** $CC$

1: randomly select $k$ data from $n$ input data, they are $N_{x_1}$, $N_{x_2}$, $N_{x_3}$, $\cdots$, $N_{x_k}$
2: Calculate the coefficients of Lagrange interpolation, according to the selected $x_i$
3: **for all** $x_i \in D_p$ **do**
4: $\quad coeff[i] = \prod_{1 \leqslant j \leqslant k, j \neq i} \frac{-x_j}{x_i - x_j}$
5: **end for**
6: $CC = Dec(\prod_{i=1}^{k} N_{x_i}^{coeff[i]})$
7: **return** $CC$

### 5.2 Aggregation Protocol for Min

It is great help to the control center to intelligent pricing and electricity distribution that mastering the peak and valley information of customers' electricity consumption. Our scheme can calculate the max and min electricity consumption of all customers.

1. Customers represent their electricity consumption data in $n$ bits. $d_{i,j}$ is the data with the size of one bit. If the electricity consumption data of the $i$-th customer is not equal to $j$, the value of $d_{i,j}$ is 0; If the electricity consumption data of the $i$-th customer is equal to $j$, the value of $d_{i,j}$ is 1. And then customers encrypt each $d_{i,j}$ using the public key $pk$.
2. Customers send the data to the edge nodes. The $i$-th customer $SM_i$ needs to send its $d_{i,j}$ to the $j$-th edge node $EN_j$. Then, each edge node calculates the product of the data they receive and transfer the calculation results to the control center.

$$N_j = \prod_{s=1}^{m} Enc(d_{s,j}) \tag{16}$$

3. The control center decrypts the results uploaded from the edge nodes in turn and get $\{Dec(N_1), Dec(N_2), Dec(N_3), \cdots, Dec(N_n)\}$. The number of the edge node corresponding to the first value that is not 0 is the minimum value of electricity consumption, and the number of the edge node corresponding to the last value that is not 0 is the maximum value of electricity consumption. The calculation method of the maximum aggregate value is similar.

### 5.3 Aggregation Protocol for Electricity Bill

In smart grid, the control center can dynamically set the electricity price, which means that the electricity price of different customers may be different at the same time. Here, we don't care about how the control center formulates the electricity price. We care about how it can get the total electricity price without infringing on the privacy of customers. If the unit price of the $i$-th customer is $w_i$, the control center can calculate $\sum_{s=1}^{m} w_s M_s$.

1. The control center broadcasts the electricity consumption unit price of all customers to all edge nodes. There are $m$ customers and we assume that the unit price of $SM_i$ is $w_i$. So, every edge receive $\{w_1, w_2, w_3, \cdots, w_m\}$.
2. Each smart meter divides the electricity consumption data into $n$ parts depending on shamir threshold scheme, and uses the public key $pk$ to encrypt the divided data. This process is the same as the aggregation protocol for summation.
3. Next, customers need to send the encrypted divided data to the edge nodes. The $j$-th divided data of each customer shall be sent to the $j$-th edge node. For example, the edge node $EN_j$ will receive encrypted split data from all customers, which should be $\{Enc(M_{1,j}),\ Enc(M_{2,j}),\ Enc(M_{3,j}),\ \cdots,\ Enc(M_{m,j})\}$.

4. Each edge node receives $m$ pieces of split data, which comes from different customers. The edge node calculates the power of each fragment and its electricity unit price, for example, $Enc(M_{1,j})^{w_1}$, $Enc(M_{2,j})^{w_2}$, $Enc(M_{3,j})^{w_3}$, etc. Then calculating the product of these split data. Specifically, the calculation result $N_j$ of the $j$-th edge node $EN_j$ should be

$$N_j = \prod_{s=1}^{m} Enc(M_{s,j})^{w_s} \tag{17}$$

According to the homomorphism of paillier cryptosystem, the power operation of scalar on ciphertext is equivalent to scalar multiplication on plaintext, and the multiplication of these encrypted split data is equivalent to their summation and then encryption. All edge nodes send their calculation results to the control center.

5. The control center randomly selects $k$ edge nodes from $n$ edge nodes. What the control center does is the same as the summation protocol. **Algorithm 1** describes this process. Finally, the result calculated by the control center is the total electricity price of the customers.

$$CC = Dec(\prod_{i=1}^{k} N_{x_i}^{\prod_{1 \leqslant j \leqslant k, j \neq i} \frac{-x_j}{x_i - x_j}}) = \sum_{s=1}^{m} w_s M_s \tag{18}$$

## 6 Security Analysis

The control center cannot get customers' real-time electricity consumption data when colludes with less than $k$ edge nodes at the same time in the proposed system. Generally, if the control center colludes with $c$ edge nodes ($c < k$) at the same time, and we express these $c$ purchased edge nodes as $\{EN_{y_1}, EN_{y_2}, EN_{y_3}, \cdots, EN_{y_c}\}$. The additional information that the control center can get from the purchased edge node $EN_{y_j}$ is the $j$-th encrypted split data from all $m$ customers $\{Enc(M_{1,j}), Enc(M_{2,j}), Enc(M_{3,j}), \cdots, Enc(M_{m,j})\}$. Then, if the control center wants to calculate the electricity consumption data $M_i$ of the $i$-th customer $SM_i$, what the control center knows is $c$ encrypted split data, and because the control center has private key $sk$, it can get $\{M_{i,y_1}, M_{i,y_2}, M_{i,y_3}, \cdots, M_{i,y_c}\}$. However, $c$ split data is not enough to restore the orignal $k-1$ degree polynomial. So, the control center cannot calculate $M_i$ using $c$ split data according to shamir threshold scheme [16]. In practice, edge nodes in the same network can be provided by different edge computing service providers. It is difficult for the control center to collude with many edge nodes at the same time.

## 7 Performance

We conduct experiments on a standard 64-bit Windows 10 system with Intel Core i7, and the proposed scheme is implemented by Python 3.7. We use the

phe library of python to implement Paillier homomorphic encryption. We use random numbers to simulate the reading of the meter, mainly measuring the time required for each part to complete the algorithm. We analyze the computational overhead of smart meters, edge nodes and the control center.

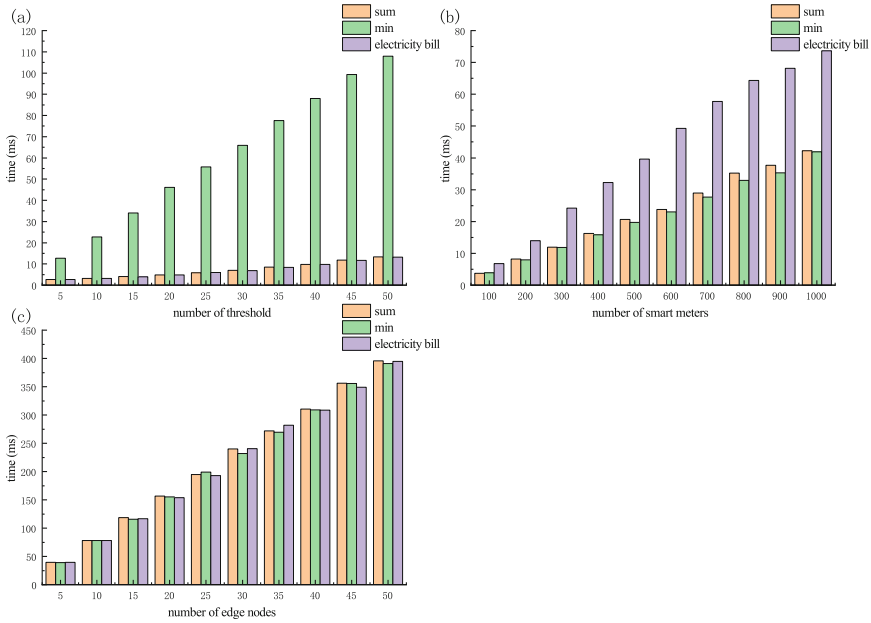



**Fig. 2.** Computational overhead of CC(a), EN(b) and SM(c).

Figure 2(a) shows the change of the calculation time(ms) of the control center with the number of threshold. We can see that the time consumption of the control center in the aggregation protocol for electricity bill and the aggregation protocol for sum is the same, because the actions performed by the control center are the same in the two aggregation protocols. For the aggregation protocol for min, the amount of calculation performed by the control center is related to the location of the minimum value. We consider the worst case, that is, the control center needs to decrypt all data. The number of data it needs to decrypt is the same as the number of edge nodes.

Figure 2(b) shows the change of the calculation time of edge nodes with the number of smart meters. Edge nodes take the longest time in the aggregation protocol for electricity bill, because compared with the other two aggregation protocols, edge nodes have to perform additional power operation on the ciphertext. The main operation of summation aggregation and minimum aggregation is multiplication on ciphertext, so the time cost is similar.

Figure 2(c) shows the change of the calculation time of smart meters with the number of edge nodes. We can see from the figure that the overhead of the

three aggregation protocols is roughly the same, because encrypted data is the main part of time consumption in the calculation of smart meter. In our three aggregation protocols, the amount of data that smart meters need to encrypt is the same as the number of edge nodes.

It can be seen that our scheme can complete the work of sum, min and calculation of electricity charge within an acceptable time. Compared with other data aggregation work, the time-consuming of this scheme is roughly the same, but it has the ability to resist collusive attacks.

## 8 Conclusion

We have proposed a collusion-tolerant data aggregation method for smart grid by leveraging Paillier homomorphic encryption and Shamir threshold scheme. In our scheme, the control center can get aggregation for sum, max/min and electricity bill. Moreover, we ensure that when edge nodes collude with the control center, the real-time power usage data of customers will not be leaked. Security analysis displays the scheme can protect customers' privacy information effectively when collusion occurs. The result of performance evaluation displays the scheme is feasible and efficient.

**Acknowledgements.** This work was supported in part by the National Nature Science Foundation of China (No. 62102429, 62102422, 62072466, 61872372), and the NUDT Grants (No. ZK19-38).

## References

1. Cai, Z., Chen, Q.: Latency-and-coverage aware data aggregation scheduling for multihop battery-free wireless networks. IEEE Trans. Wirel. Commun. **20**(3), 1770–1784 (2021)
2. Cheng, S., Cai, Z., Li, J., Gao, H.: Extracting kernel dataset from big sensory data in wireless sensor networks. IEEE Trans. Knowl. Data Eng. **29**(4), 813–827 (2017)
3. Gai, N., Xue, K., He, P., Zhu, B., Liu, J., He, D.: An efficient data aggregation scheme with local differential privacy in smart grid. In: 2020 16th International Conference on Mobility, Sensing and Networking (MSN), pp. 73–80 (2020)
4. Gong, Y., Cai, Y., Guo, Y., Fang, Y.: A privacy-preserving scheme for incentive-based demand response in the smart grid. IEEE Trans. Smart Grid **7**(3), 1304–1313 (2016)
5. Gope, P., Sikdar, B.: A lightweight and privacy-preserving data aggregation for dynamic pricing-based billing in smart grids. In: 2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), pp. 1–7 (2018)
6. Guan, Z., et al.: APPA: an anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT. J. Netw. Comput. Appl. **125**, 82–92 (2019)
7. He, Z., Cai, Z., Cheng, S., Wang, X.: Approximate aggregation for tracking quantiles and range countings in wireless sensor networks. Theor. Comput. Sci. **607**, 381–390 (2015)

8. Liu, J.N., Weng, J., Yang, A., Chen, Y., Lin, X.: Enabling efficient and privacy-preserving aggregation communication and function query for fog computing-based smart grid. IEEE Trans. Smart Grid **11**(1), 247–257 (2020)
9. Lu, W., Ren, Z., Xu, J., Chen, S.: Edge blockchain assisted lightweight privacy-preserving data aggregation for smart grid. IEEE Trans. Netw. Service Manage. **18**(2), 1246–1259 (2021)
10. Luo, X., Xue, K., Xu, J., Sun, Q., Zhang, Y.: Blockchain based secure data aggregation and distributed power dispatching for microgrids. IEEE Trans. Smart Grid **12**(6), 5268–5279 (2021)
11. Lyu, L., Nandakumar, K., Rubinstein, B., Jin, J., Bedo, J., Palaniswami, M.: PPFA: privacy preserving fog-enabled aggregation in smart grid. IEEE Trans. Industr. Inform. **14**(8), 3733–3744 (2018)
12. Ni, J., Zhang, K., Lin, X., Shen, X.S.: Balancing security and efficiency for smart metering against misbehaving collectors. IEEE Trans. Smart Grid **10**(2), 1225–1236 (2019)
13. Okay, F.Y., Ozdemir, S.: A fog computing based smart grid model. In: 2016 International Symposium on Networks, Computers and Communications (ISNCC), pp. 1–6 (2016)
14. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
15. Saleem, A., et al.: FESDA: fog-enabled secure data aggregation in smart grid IoT network. IEEE Internet Things J. **7**(7), 6132–6142 (2020)
16. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
17. Wang, H., Wang, Z., Domingo-Ferrer, J.: Anonymous and secure aggregation scheme in fog-based public cloud computing. Future Gener. Comput. Syst. **78**, 712–719 (2017)
18. Wang, J., Wu, L., Zeadally, S., Khan, M.K., He, D.: Privacy-preserving data aggregation against malicious data mining attack for IoT-enabled smart grid. ACM Trans. Sen. Netw. **17**(3), 1–25 (2021)
19. Wang, P., Liu, S., Ye, F., Chen, X.: A fog-based architecture and programming model for IoT applications in the smart grid. arXiv preprint arXiv:1804.01239 (2018)
20. Wu, L., Xu, M., Fu, S., Luo, Y., Wei, Y.: FPDA: fault-tolerant and privacy-enhanced data aggregation scheme in fog-assisted smart grid. IEEE Internet Things J. **9**(7), 5254–5265 (2021)
21. Zahoor, S., Javaid, N., Khan, A., Ruqia, B., Muhammad, F.J., Zahid, M.: A cloud-fog-based smart grid model for efficient resource utilization. In: 2018 14th International Wireless Communications Mobile Computing Conference (IWCMC), pp. 1154–1160 (2018)
22. Zhao, S., et al.: Smart and practical privacy-preserving data aggregation for fog-based smart grids. IEEE Trans. Inf. Forensics Secur. **16**, 521–536 (2021)

# Network Defense Resource Allocation Scheme with Multi-armed Bandits

Ning Huang, Xue-cai Feng, Rui Zhang, Xiu-gui Yang, and Hui Xia(✉)

College of Computer Science and Technology, Ocean University of China, Qingdao 266100, China
xiahui@ouc.edu.cn

**Abstract.** The problem of limited defense resources owned by the network platform needs to be solved by designing a reasonable defense resource allocation scheme in Industrial Internet of Things (IIoT). However, most of the previously studied defense resource allocation schemes do not consider the impact of network cheat on the defender's total expected utility, resulting in the defender's total expected utility not being optimal. To address this problem, this paper proposes a network defense resource allocation scheme with multi-armed bandits (NDRAS) to maximize the defender's total expected utility. The scheme first proposes a random generation method of node shell configuration based on network cheat, by considering the impact of network cheat on the defender's total expected utility, masking information about the real configuration of nodes, to increase the uncertainty of the attacker's attack on each node and thus reduce the likelihood of the attacker's success. Subsequently, the decomposability and Lipschitz continuity of the defender's total expected utility is exploited to reduce the gap between the cumulative discrete optimal benefit and the continuous optimal benefit, to maximize the defender's total expected utility and thus make the defender's total expected utility optimal. Finally, the detailed experimental results confirm the effectiveness of NDRAS, indicating that the new scheme can give a reasonable defense resource allocation scheme to maximize the defender's total expected utility.

**Keywords:** Industrial Internet of Things · Multi-armed bandits · Network cheat · Defense resource allocation

## 1 Introduction

Industrial Internet of Things (IIoT) is an important driver of intelligent change in the global industrial system [1], playing an important role in traditional manufacturing, energy, transportation [2], and other fields, but its network platform

Supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62172377 and 61872205, and the Natural Science Foundation of Shandong Province under Grant No. ZR2019MF018.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 326–337, 2022.
https://doi.org/10.1007/978-3-031-19208-1_27

has limited defense resources, resulting in network nodes or edge devices lacking high-performance security protection against various types of network attacks, making them highly vulnerable to various malicious codes and hacker attacks, posing serious security risks to network end-users. For example, in 2021, JBS, the world's largest meat processor, suffered a cyber attack that left the plant unable to function and forced workers to shut down, placing a severe financial burden on the company and its employees. In the same year, Colonial Pipeline, the largest fuel pipeline operator in the US, was hit by a ransomware attack that shut down its energy supply network, causing the first national emergency in the US due to a cyber attack and causing a significant negative impact on the country's energy supply. Therefore, improving network security with limited network defense resources is an urgent problem that needs to be addressed. A well-designed defense resource allocation scheme can effectively solve this problem.

The defense resource allocation schemes that exist are mainly divided into target-based attack and defense game schemes [3,4] and network structure-based attack and defense game schemes [5–7]. Target-based attack and defense game schemes are designed to focus the attacker (defender) on some individual target of the attack (defense), but such defense schemes are not applicable in realistic attack and defense game situations. Network structure-based attack and defense game schemes in which the attackers and defenders rely on the network structure for strategy selection, mostly seek to maximize the sum of the expected utilities of each node, and are more in line with realistic attack and defense game situations than target-based attack and defense game schemes. Depending on the application scenario, the network structure-based attack and defense game schemes can be further divided into two main categories: benefits of nodes known to defenders and benefits of nodes unknown to defenders. Among them, the scheme in which defenders know the benefits of nodes can obtain the optimal defense strategy according to its proposed game model, but this type of allocation scheme does not consider the situation in which the cumulative discrete optimal benefit (defender) approaches the continuous optimal benefit (defender) without knowing the benefits of each node, resulting in this type of scheme being suitable only for application scenarios in which defenders know the distribution of the benefit function of each node. The scheme in which defenders do not know the benefits of nodes solves the situation in which defenders do not know the distribution of the benefit function [8–10], but such defense resource allocation schemes do not take into account the factors influencing the defender's total expected utility, resulting in the defender's total expected utility not being optimal.

To address the above problems, this paper proposes a scheme for allocating network defense resources using multi-armed bandits by reasonably modeling the benefit function while considering the impact of the factor of network cheat on the defender's total expected utility [11–13]. As far as we know, this scheme is the first defense resource allocation scheme based on network cheat applied to

network defense resource allocation. The main contributions of this paper are as follows.

- To reduce the likelihood of successful attacks by attackers, this paper proposes a random generation method of node shell configuration based on network cheat. The method confuses the attacker's perception of the true configuration of each node by considering the impact of network cheat as an influencing factor on the defender's total expected utility to add a shell configuration to each node that is different from its importance.
- To optimize the defender's total expected utility, this paper uses the decomposability and Lipschitz continuity of the defender's total expected utility function to maximize the sum of the defender's expected utility of each node, thus reducing the gap between the cumulative discrete optimal utility and the continuous optimal utility.
- To verify the effectiveness of the above scheme, this paper firstly evaluates NDRAS with the help of the metric of total defender expected utility. The experimental results show that, given other assumptions, the total defender expected utility for NDRAS increases as the total number of defense resources owned by the network platform increases, and the optimal total defender expected utility for NDRAS decreases as the number of network nodes increases. Secondly, this paper compares NDRAS with NDRAS-NOD, CUCB, Zooming, and Lizard with the help of two metrics, total defender expected utility and regret, and the experimental results show that, given other assumptions, the total defender expected utility for NDRAS shows a relatively good upward trend over time, and the regret for NDRAS shows a downward trend over time.

## 2 Problem Definition

### 2.1 Problem Description

We model the defender and attacker based on the Stackelberg Security Game (SSG) model, where both are strategic players. In the SSG, the defender is the leader and determines its strategy first, and the attacker is the follower and makes its own best response after viewing the defender's strategy. In this paper, the network is modeled as an undirected graph consisting of a set of nodes $V = \{1, 2, ..., n\}$ and a set of node-connected edges $E$. The set of all nodes in the network is $V$, and each node has its real configuration. The total number of defense resources owned by the network platform is *budget*. In the Combinatorial Multi-Arm bandit (CMAB) problem, the optimal arm is chosen to maximize the combinatorial expected utility, where each arm has its defender benefit and attacker benefit, i.e., each arm has a different expected utility, where the optimal arm is a subset of all the arms. We define a CMAB problem where each node in the network is analogous to an arm of the MAB, i.e., there are $n$ arms, and the defender obtains the best strategy in a trade-off between exploration and exploitation. Unlike the classical setting of MABs, we give the distribution

of benefits of each node in the network through the rewards and penalties of the attacker and defender, and adds a shell configuration to each node that is different from the real configuration of nodes, and obtains the probability of each node being attacked through the SUQR model.

### 2.2 Attacker Strategies

The heterogeneity of nodes in a complex network makes nodes differ in importance and therefore the true configuration of each node. The greater the true configuration of a node, the more important the node is, i.e., the greater the node's need for defense resources. The importance of a node should form a linear logical relationship with its true configuration.

For the structural specificity of complex networks, this paper integrates the characteristics of local and global networks. For $\forall i \in V$, let $\mathbf{y}_i = \{y_1, y_2, ...., y_t\}$ denote the set of features of node $i$. Based on the nature of semi-local centrality, this paper relies on the entropy method to assess the importance of nodes by their features and thus obtain the true configuration of nodes. For $\forall i \in V$, let $TC_i$ denote the critical value of the number of defense resources required to fully protect node $i$ from damage, i.e., the true configuration of node $i$. Then the true configuration of all nodes in the network can be expressed as $\mathbf{TC} = \{TC_1, TC_2, ..., TC_n\}$, which satisfies $\sum_{i=1}^{n} TC_i > budget$. Let $\mathbf{DN} = \{DN_1, DN_2, ..., DN_n\}$ denote the number of defense resources allocated to each node. When the attacker launches an attack, for $\forall i \in V$, if $DN_i \geq TC_i$, the defender (attacker) will receive the corresponding reward $R_i^d > 0$ (punishment $P_i^a > 0$); if $DN_i < TC_i$, the defender (attacker) will receive the corresponding punishment $P_i^d > 0$ (reward $R_i^a > 0$). When the attacker does not launch an attack, for $\forall i \in V$, node $i$ is fully protected regardless of whether the number of defense resources allocated to node $i$ is greater than its true configuration, and the defender will receive the corresponding reward $R_i^d > 0$. Due to the nature of the Stackelberg game, for $\forall i \in V$, $R_i^d > P_i^d$ and $R_i^a > P_i^a$ need to be satisfied, and for ease of analysis, let $R_i^d = R_i^a$, $P_i^d = P_i^a$. To motivate the defender to allocate more defense resources to the core nodes, the reward function $R$ and the punishment function $P$ should form a non-linear and increasing logical relationship with the true configuration of nodes. Therefore, in this paper, the reward function $R$ (punishment function $P$) is set as a class of monotonically increasing concave functions.

The attacker's benefit $v^a$, which is related to the reward and punishment of the node, the number of defense resources allocated to the node, and the true configuration of the node, then for $\forall i \in V$, the attacker's benefit function is:

$$v_i^a = -\min\{\left\lfloor \frac{DN_i}{TC_i} \right\rfloor, 1\} \cdot P_i^a + \max\{1 - \left\lfloor \frac{DN_i}{TC_i} \right\rfloor, 0\} \cdot R_i^a \tag{1}$$

Network cheat is one of the key methods of misleading an attacker by hiding or providing inaccurate system information. The purpose of network cheat is to mislead the attacker into attacking non-core nodes by hiding or misrepresenting the configuration of nodes in the network.

This paper introduces a random generation method of node shell configuration based on network cheat. It is assumed that the attacker is unable to distinguish between nodes with the same shell configuration. Let $M = \{1, 2, ..., m\}$ denote the set of node shell configurations in the network that satisfies $m < n$. Then the shell configurations of all nodes in the network can be expressed as $\mathbf{OC} = \{OC_1, OC_2, ...OC_m\}$. This paper randomly assigns the shell configuration of each node in the network. Given an integer matrix $\mathbf{\Phi}$ of size $n \times m$ represents the correspondence between nodes and their shell configurations. Where, for $\forall i \in V$ and $\forall j \in M$, $\Phi_{ij}$ represents the number of nodes with a real configuration of $TC_i$ and a shell configuration of $OC_j$, satisfying $\Phi_{ij} \in \{0, 1\}$. Therefore, for $\forall j \in M$, the probability that the attacker attacks a node with a shell configuration of $OC_j$ is:

$$P_{OC_j} = \frac{\Phi_{1j} v_1^a + \Phi_{2j} v_2^a + ...\Phi_{nj} v_n^a}{\sum_{i \in V} v_i^a} = \frac{\sum_{i \in V} \Phi_{ij} v_i^a}{\sum_{i \in V} v_i^a} \tag{2}$$

The attacker's strategy depends on the attacker's benefit and the attacking node shell configuration's probability. In this paper, the probability of each node in the network being attacked, i.e., the attacker's strategy, is obtained by the SUQR model. Thus, for $\forall i \in V$, the probability of node $i$ being attacked is:

$$q_i = \frac{e^{\lambda(\sum_{j \in M} \Phi_{ij} P_{OC_j}) + \omega v_i^a}}{\sum_{k \in V} e^{\lambda(\sum_{l \in M} \Phi_{kl} P_{OC_l}) + \omega v_k^a}} \tag{3}$$

where $\lambda$ and $\omega$ are regulation parameters.

### 2.3 Defender Strategies

Given a discrete spacing *space*, divide the total number of defense resources *budget* into $J$ copies according to *space* increasing from zero, i.e., $\mathbf{P} = \{p_1, ..., p_J\}$, where the set of resource copies can be expressed as $NP = \{1, 2, ..., J\}$. Given an integer matrix $\mathbf{\Omega}$ of size $n \times J$ represents whether to allocate the number of defense resources to each node in the network. Where, for $\forall i \in V$, $\mathbf{\Omega_i} = \{\Omega_{i,1}, ..., \Omega_{i,J}\}$ represents whether node $i$ is allocated the number of defense resources. For $\forall i \in V$ and $\forall j \in NP$, $\Omega_{i,j}$ represents the resource copy $p_j$ allocated to node $i$, satisfying $\Omega_{i,j} \in \{0, 1\}$.

The number of defense resources allocated to each node in the network is known to be $\mathbf{DN} = \{DN_1, DN_2, ..., DN_n\}$, i.e., the defender's strategy. Then for $\forall i \in V$, the number of defense resources allocated to node $i$ is:

$$DN_i = \sum\nolimits_{j \in NP} \Omega_{i,j} \cdot p_j \tag{4}$$

The defender's benefit $v^d$, which is related to the reward and punishment of the node, the number of defense resources allocated to the node, and the true configuration of the node, then for $\forall i \in V$, the defender's benefit is:

$$v_i^d = \min\{\left\lfloor \frac{DN_i}{TC_i} \right\rfloor, 1\} \cdot R_i^d - \max\{1 - \left\lfloor \frac{DN_i}{TC_i} \right\rfloor, 0\} \cdot P_i^d \tag{5}$$

When the attacker launches an attack, the defender gets $v^d$ and the attacker gets $v^a$; conversely, the defender gets $R^d$. Hence, with reasonable modeling of the node's benefit function and considering the factors influencing the defender's expected utility, for $\forall i \in V$, the defender's expected utility function is:

$$EU_i^d = q_i \cdot v_i^d + (1 - q_i) \cdot R_i^d \tag{6}$$

Therefore, the defender's total expected utility function is:

$$TE^d = \sum\nolimits_{i=1}^{n} EU_i^d = \sum\nolimits_{i=1}^{n} (q_i \cdot v_i^d + (1 - q_i) \cdot R_i^d) \tag{7}$$

The measurement between the set of arms for which the non-optimal defender's expected utility is chosen and the set of arms for which the optimal defender's expected utility is chosen is central to online learning, provided that the benefits of each node are unknown to defenders. Where the gap between the defender's total expected utility for these two arm sets is regret. Let *Optimal* denote the optimal defender's total expected utility when choosing the set of arms with the optimal expected utility. The objective of this paper is to minimize regret over a time horizon $T$ to obtain the defense strategy when the defender's total expected utility is maximized.

## 3 Defense Resource Allocation Strategy

### 3.1 Characteristics of the Defender's Total Expected Utility

The defender's total expected utility function is the sum of the defender's expected utility across the nodes in the network, i.e., the defender's total expected utility function has decomposability. The defender's total expected utility function depends on the defender's and attacker's benefit functions, and also on the relationship between the true configuration of nodes and the size of the number of defense resources allocated to nodes. Based on the features of nodes we can obtain the importance of each node in the network and thus the true configuration of nodes. Assume that the defender's expected utility function satisfies Lipschitz continuity in terms of the features of nodes and the number of defense resources allocated to nodes. Then two different nodes in the network with the same features and the same number of defense resources will have the same defender's expected utility.

Thus, the defender's total expected utility function has decomposability and Lipschitz continuity.

### 3.2 NDRAS

**Upper Confidence Bound.** With the benefits of each node unknown to defenders, we formulate the defender's problem as a CMAB problem and uses the Upper Confidence Bound (UCB) algorithm to solve it. In the CMAB problem,

**Algorithm 1.** Defense Strategy

**Input**: An undirected graph $G(V, E)$, a time horizon $T$, the total number of defense resources owned by the network platform *budget*, real configuration of nodes **TC**, shell configuration of nodes **OC**, discrete interval *space*
**Parameter**: $comTE_0^d(i) = 0$,$comN_0(i,j) = 0$, $\forall i \in N, \forall j \in NP$

```
for t = 1 to T do
    Select the optimal arm using Eq. (10)
    Compute the defender expected utility EU_i^d, ∀i ∈ V using Eq. (6)
    for i = 1 to n do
        comTE_t^d(i) += EU_i^d
        for j = 1 to J do
            comN_t(i,j) += 1
        end for
    end for
end for
```

the current optimal arm is selected at each moment in time based on historical observations to obtain the combinatorial utility.

For $\forall i \in V$ and $\forall j \in NP$, in time step $t$, let $Avg\_Reward_t(i,j) = \frac{comTE_t^d(i)}{comN_t(i,j)}$ denote the defender's average expected utility at node $i$, where $comTE_t^d(i)$ is the defender's cumulative expected utility and $comN_t(i,j)$ is the cumulative number of arm pulls. The defender's cumulative expected utility as well as the cumulative number of arm pulls of each node in the network should be initialized to zero to satisfy the definition of online learning. Thus, for $\forall i \in V$, the UCB based only on node $i$'s own observations in time step $t$ is:

$$obsUCB_t(i,j) = Avg\_Reward_t(i,j) + r_t(i,j) \tag{8}$$

where $r_t(i,j) = \sqrt{\frac{3\log(t)}{2comN_t(i,j)}}$ is the degree of uncertainty in the selection of node $i$.

The defender's total expected utility is known to have Lipschitz continuity. Then for $\forall i \in V$, its UCB can further explore the Lipschitz continuity between arms, i.e., by adding the distance function of arm $i$ from all other arms. This distance function depends on the similarity of the node features and the similarity of the number of defense resources allocated to the node. Thus, for $\forall i \in V$, the UCB of node $i$ in time step $t$ is:

$$UCB_t(i,j) = \min_{m \in V, n \in NP} \{obsUCB_t(m,n) + L \cdot (\max\{0, p_n - p_j\} + D(\mathbf{y}_i, \mathbf{y}_m))\} \tag{9}$$

where $L$ is the Lipschitz constant.

**The Optimal Arm Selection.** In this paper, the selection of the optimal arm is the choice of the number of defense resources issued to the nodes by the network platform. The optimal arm can be obtained by calculating the maximum

value of the sum of the UCBs of all arms, which gives the number of defense resources allocated to each node.

The problem can be reduced to a linear programming mathematical model of finding an optimal solution to a constrained objective function, where the objective function is the maximum of the sum of the UCBs of all arms. Then, at time step $t$, the linear programming $L$ is:

$$\begin{aligned} &\max_{\mathbf{\Omega}} \sum_{i \in V} \sum_{j \in NP} \Omega_{i,j} \cdot UCB_t(i,j) \\ &s.t. \, \Omega_{i,j} \in \{0,1\}, \forall i \in V, j \in NP \\ &\sum_{i \in V} \Omega_{i,j} = 1, \forall j \in NP \\ &\sum_{i \in V} \sum_{j \in J} \Omega_{i,j} \cdot p_j \leq budget \end{aligned} \tag{10}$$

The objective of this paper is to obtain the defense strategy when the defender's total expected utility is maximized to allocate the limited defense resources to each node in the network. The number of defense resources allocated to each node, i.e., the defense strategy, can be obtained by linear programming. The detailed solution process is shown in Algorithm 1.

## 4 Experiment

In this section, we verify the factors influencing the defender's total expected utility and verify that NDRAS (Algorithm 1) yields superior experimental results. In the experiment, firstly, one metric, the defender's total expected utility, is introduced to evaluate the algorithm NDRAS proposed in this paper. Secondly, two metrics, the defender's total expected utility and regret, are introduced to compare the algorithm NDRAS proposed in this paper with other algorithms.

### 4.1 Experimental Setup

The experiments are attack and defense game interactions in a LAN with $n = 5$, 10, or 20 nodes, and the results are the average of 30 experiments. Let $fullP$ denote the total number of defense resources required when all nodes in the network are fully protected, i.e., $fullP = \sum_{i=1}^{n} TC_i$, which satisfies $budget < fullP$. We set the *space* to $\frac{2 \cdot budget}{n \cdot (n-1)}$. We use the three metrics of *k-shell* values, clustering coefficients, and the smallest eigenvalue of grounded laplacian matrices of nodes as features. For $\forall i \in V$, let $I_i$ denote the importance of node $i$. The importance of a node can be obtained based on its features, and the importance of a node forms a linear logical relationship with its true configuration. We map the real configuration of nodes to the range of $(0, 1]$. Assuming that node $a$ is the most important in the network, the true configuration of node $i$ is $TC_i = \frac{I_i}{I_a} * TC_a$ for $\forall i \in V$, when the true configuration of node $a$ is $TC_a \in \left[\frac{n-1}{n}, 1\right]$. We set the reward function $R$ and the punishment function $P$ to be a class of monotonically increasing concave functions, then for $\forall i \in V$, let $R_i^d = R_i^a = e^{TC_i} - 1$, $P_i^d = P_i^a = e^{TC_i} - TC_i - 1$. At the same time, given a time horizon $T = 500$, we verify that Algorithm 1 can achieve effective iterations.

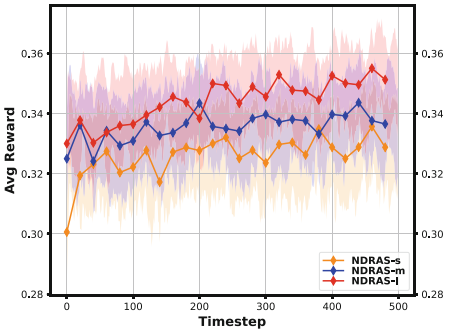



**Fig. 1.** Comparison of the defender's total expected utility at different *budget*

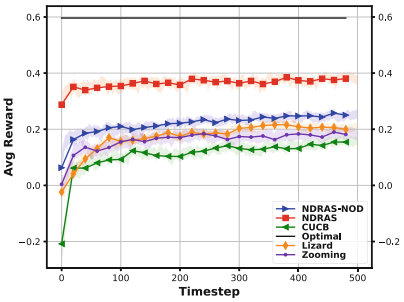
### 4.2 Evaluate NDRAS

*The number of defense resources owned by the network platform budget:* In Fig. 1, we examine the comparison of the defender's total expected utility for NDRAS at different *budget*. We set $n = 10$ and $T = 500$. We can see that there is an overall upward trend in the defender's total expected utility as time moves from 0 to 500. We can also see that the defender's total expected utility is higher for NDRAS-l ($budget = 4/5 * fullP$) than for NDRAS-m ($budget = 3/4 * fullP$), and the defender's total expected utility is higher for NDRAS-m ($budget = 3/4 * fullP$) than for NDRAS-s ($budget = 2/3 * fullP$), because the higher the number of defense resources owned by the network platform, the higher the defender's total expected utility.

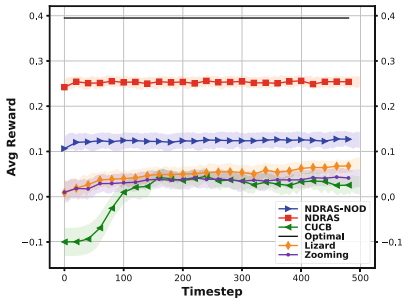
*The number of network nodes n:* In Fig. 2 and Fig. 4(a), we examine the comparison of the defender's total expected utility for NDRAS at different $n$. We set $budget = 3/4 * fullP$ and $T = 500$. In Fig. 2(a), Fig. 4(a), and Fig. 2(b), we can see that the overall trend of the defender's total expected utility increases with time for $n = 5$, 10, or 20 nodes in the network. We can also see that *Optimal* decreases as the number of network nodes increases.

### 4.3 Experimental Comparison of NDRAS with Other Algorithms

In our experiments, we compare the proposed NDRAS with three algorithms, CUCB [12], Zooming [13], and Lizard [6], and show that NDRAS has good performance. CUCB considers the possibility of more base arms being triggered on the set of arms that have already been triggered, but does not consider the similarity between the two arms. Zooming combines the upper confidence bounds used in the UCB-1 algorithm with adaptive discretization, but does not consider the decomposability of the benefit function. Lizard uses historical data as the initial input and considers the similarity between the two arms and the decomposability of the benefit function. Our algorithm does not use historical data, but sets up a reasonable form of the benefit function based on the Lizard algorithm, while considering network cheat as an influencing factor, to obtain a defense strategy when the defender's total expected utility is maximized.

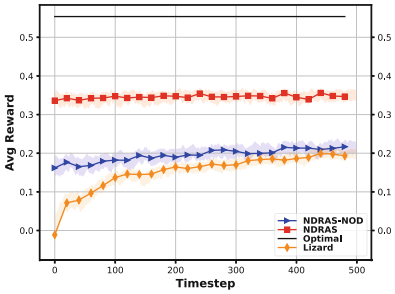



(a) $n = 5$

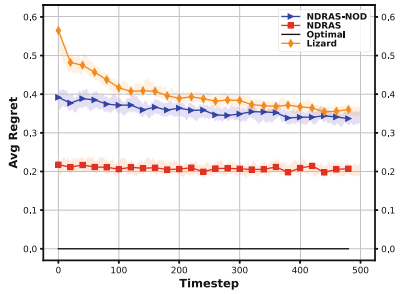



(b) $n = 20$

**Fig. 2.** Comparison of the defender's total expected utility at different $n$




(a) The defender's total expected utility



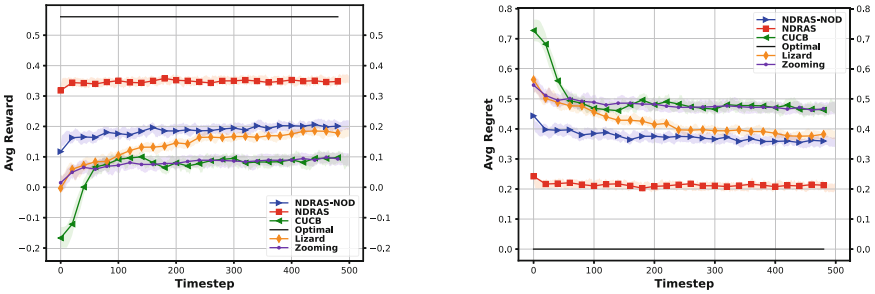

(b) Regret

**Fig. 3.** Comparison of NDRAS, NDRAS-NOD and Lizard

### 4.4 Influencing Factors

*Network cheat:* In Fig. 3, we examine the impact of network cheat on the defender's total expected utility as well as regret. Among these, NDRAS-NOD is the case where NDRAS does not take into account network cheat. We set $n = 10$, $budget = 3/4 * fullP$, and $T = 500$. We can see that in Fig. 3(a), the defender's total expected utility for NDRAS is significantly higher than that for NDRAS-NOD, and in Fig. 3(b), the regret for NDRAS is significantly smaller than that for NDRAS-NOD. This is because, in NDRAS, we consider network cheat as an influencing factor, indicating that network cheat can confuse the attacker to reduce the likelihood of a successful attack.

*Attacker behavior:* In Fig. 3, we examine the impact of attacker behavior on the defender's total expected utility as well as regret. Among these, NDRAS-NOD is the case where Lizard does not consider attacker behavior. We set $n = 10$, $budget = 3/4 * fullP$, and $T = 500$. We can see that in Fig. 3(a), the




(a) The defender's total expected utility (b) Regret

**Fig. 4.** Comparison of NDRAS, NDRAS-NOD, CUCB, Zooming, and Lizard

defender's total expected utility for NDRAS-NOD is higher than that for Lizard, and in Fig. 3(b), the regret for NDRAS-NOD is significantly smaller than that for Lizard, because in NDRAS-NOD we consider attacker behavior as an influencing factor.

### 4.5 Comparison Results

In Fig. 4, we examine the experimental performance comparison of the defender's total expected utility as well as regret for the NDRAS, NDRAS-NOD, CUCB, Zooming, and Lizard algorithms. We set $n = 10$, $budget = 3/4 * fullP$, $T = 500$. In both Fig. 4(a) and Fig. 4(c), we can see that the overall trend of the defender's expected utility is upward as time increases from 0 to 500. We can also see that NDRAS and NDRAS-NOD show a relatively good upward trend, because in NDRAS-NOD we consider attacker behavior as an influencing factor, and in NDRAS we consider network cheat as an influencing factor. Similarly, in Fig. 4(b) and Fig. 4(d), we can see that the overall trend of regret decreases with increasing time, where NDRAS presents the smallest regret.

## 5 Conclusion

We propose a network defense resource allocation scheme with multi-armed bandits. In this scheme, we model the attacker and defender based on the SSG model, while introducing network cheat and rationalizing the form of the benefit function on the premise that the benefits of each node are unknown to defenders and the total defense resources are limited. In our experiments, we compare the performance of the proposed algorithm with CUCB, Zooming, and Lizard. The experimental results show that our proposed algorithm maximizes the defender's total expected utility under the same experimental setup. We believe that the study in this paper will be more effectively defense against all kinds of cyber attacks on IIoT, thus effectively enhancing the overall security of the network and thus promoting the quality growth of the industrial system.

**Acknowledgements.** This research is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62172377 and 61872205, and the Natural Science Foundation of Shandong Province under Grant No. ZR2019MF018.

## References

1. Zhifang, G., Chen, H., Pingping, X., Li, Y., Vucetic, B.: Physical layer authentication for non-coherent massive SIMO-enabled industrial IoT communications. IEEE Trans. Inf. Forensics Secur. **15**, 3722–3733 (2020)
2. Zheng, C., Fan, X., Wang, C., Qi, J.: GMAN: a graph multi-attention network for traffic prediction. Proc. AAAI Conf. Artif. Intell. **34**, 1234–1241 (2020)
3. Wang, Y., et al.: Deep reinforcement learning for green security games with real-time information. Proc. AAAI Conf. Artif. Intell. **33**, 1401–1408 (2019)
4. Macke, W., Mirsky, R., Stone, P.: Expected value of communication for planning in ad hoc teamwork. Proc. AAAI Conf. Artif. Intell. **35**, 11290–11298 (2021)
5. Zhang, Y., Guo, Q., An, B., Tran-Thanh, L., Jennings, N.R.: Optimal interdiction of urban criminals with the aid of real-time information. Proc. AAAI Conf. Artif. Intell. **33**, 1262–1269 (2019)
6. Li, M., Tran-Thanh, L., Xiaowei, W.: Defending with shared resources on a network. Proc. AAAI Conf. Artif. Intell. **34**, 2111–2118 (2020)
7. Shen, W., Chen, W., Huang, T., Singh, R., Fang, F.: When to follow the tip: security games with strategic informants. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (2020)
8. Kleinberg, R., Slivkins, A., Upfal, E.: Bandits and experts in metric spaces. J. ACM **66**(4), 1–77 (2019)
9. Chen, W., Wang, Y., Yuan, Y., Wang, Q.: Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. J. Mach. Learn. Res. **17**(1), 1746–1778 (2016)
10. Lily, X., Bondi, E., Fang, F., Perrault, A., Wang, K., Tambe, M.: Dual-mandate patrols: multi-armed bandits for green security. Proc. AAAI Conf. Artif. Intell. **35**, 14974–14982 (2021)
11. Cai, Z., He, Z., Guan, X., Li, Y.: Collective data-sanitization for preventing sensitive information inference attacks in social networks. IEEE Trans. Dependable Secure Comput. **15**(4), 577–590 (2016)
12. Cai, Z., Zheng, X.: A private and efficient mechanism for data uploading in smart cyber-physical systems. IEEE Trans. Netw. Sci. Eng. **7**(2), 766–775 (2018)
13. Ye, D., Zhu, T., Shen, S., Zhou, W.: A differentially private game theoretic approach for deceiving cyber adversaries. IEEE Trans. Inf. Forensics Secur. **16**, 569–584 (2020)

# FLFHNN: An Efficient and Flexible Vertical Federated Learning Framework for Heterogeneous Neural Network

Han Sun[1,2], Yan Zhang[1,2(✉)], Mingxuan Li[1,2], and Zhen Xu[1]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{sunhan,zhangyan,limingxuan2,xuzhen}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

**Abstract.** The emergence of vertical federated learning (VFL) solves the problem of joint modeling between participants sharing the same ID space and different feature spaces. Privacy-preserving (PP) VFL is challenging because complete sets of labels and features are not owned by the same entity, and more frequent and direct interactions are required between participants. The existing VFL PP schemes are often limited by the communication cost, the model types supported, and the number of participants. We propose FLFHNN, a novel PP framework for heterogeneous neural networks based on CKKS fully homomorphic encryption (FHE). Combining the advantages of FHE in supporting types of ciphertext calculation, FLFHNN eliminates the limitation that the algorithm only supports limited generalized linear models and realizes "short link" communication between participants, and adopts the training and inference of encrypted state to ensure confidentiality of the shared information while solving the problem of potential leakage from the aggregated values of federated learning. In addition, FLFHNN supports flexible expansion to multi-party scenarios, and its algorithm adapts according to the number of participants. Our analysis and experiments demonstrate that compared with Paillier based scheme, FLFHNN significantly reduces the communication cost of the system on the premise of retaining the accuracy of the model, and the required interactions and information transmission for training are reduced by almost 2/3 and more than 30% respectively, which is suited to large-scale internet of things scenarios.

**Keywords:** Privacy-preserving · Machine learning · Vertical federated learning · Homomorphic encryption · Heterogeneous neural network

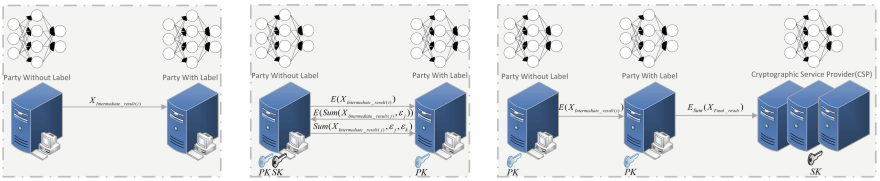

L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 338–350, 2022.
https://doi.org/10.1007/978-3-031-19208-1_28

## 1 Introduction

As a widely used privacy-preserving (PP) machine learning (ML) scheme to train neural networks (NN) [1] in a distributed manner, federated learning (FL) can achieve the purpose of joint modeling with the help of other party's data on the premise that the data are not out of the local, realizing the data "available but invisible" to a certain extent. FL was mainly applied in the horizontal distribution of data scenario when it was first proposed [2–4], horizontal federated learning (HFL). In vertical federated learning (VFL), the data is vertically distributed, and the participants hold the datasets with the same ID space and different feature spaces [5]. Participants need frequent interactions of intermediate training results to expand the feature space to enhance the model and complete the training due to the datasets without complete feature spaces and labels.

Previous studies have shown that the local data could be inferred from shared intermediate results (e.g., gradients) [6–10]. In subsequent works, cryptographic techniques were introduced, such as secure multiparty computation (SMC) [11], differential privacy (DP) [12,13], and homomorphic encryption (HE) [14,15]. The privacy of shared information is protected by blind processing of encrypted information, and in most VFL frameworks, a trusted third party is added to ensure the privacy and data security of data providers. However, these schemes generally face problems that still cannot be solved: Firstly, Frequent peer-to-peer communication between participants causes lots of communication costs and makes the system vulnerable to communication failures. Secondly, They apply only to limited generalized linear models (GLM) and always train non-linear ML models such as logical regression by seeking approximation algorithms, which inevitably affects the accuracy of models. Finally, Parties receiving the aggregated values of decryption parameters or gradients can infer information about other parties' data, resulting in the leakage of the aggregated values. Recently, a series of VFL schemes without a third party responsible for key management and decryption has been proposed [17–19]. For example, some methods allow both participants to generate key pairs and send public keys to each other to encrypt shared intermediate information. When final results need to be decrypted, noise is added to protect the confidentiality of their own information. The party who finally obtains decryption results can obtain accurate information by removing the noise added by itself. Although these methods can ensure the accuracy of the results and eliminate the limitation of the algorithm, they sharply aggravate the communication cost and computational complexity of the system (Fig. 1).

In this paper, we propose a VFL framework (FLFHNN) with fully homomorphic encryption (FHE) and a trusted cryptographic service provider (CSP), which provides confidential protection for the data and models in training and inference of heterogeneous neural networks (HNN). This framework realizes "short link" communication between participants, that is, all participants can complete the prediction task through only one communication, and eliminates the limitation of traditional schemes with a third party in supporting algorithms. In addition, the algorithm of the framework adapts according to the number of




(a) VFL without a third party and privacy protection (b) Paillier based VFL without a third party (c) CKKS based VFL with a CSP

**Fig. 1.** Different VFL schemes for HNN with various types of encryption schemes

participants. In the two-party scenario, using the training and inferential mode of encrypted state, Only CSP can obtain the plaintexts of joint training and prediction results, which can protect the shared information in FL and prevent the potential leakage of aggregated information. In three-party and above scenarios, due to the robustness of the increase in the number of participants to the leakage of aggregated value, the aggregated ciphertexts are sent to CSP for decryption before calculating the activation functions of the models. Then the aggregated information executes non-linear calculation in plaintext but in a PP manner. Specifically, our contributions are summarized as follows:

- We propose a novel PP, efficient, FHE based scheme for VFL, FLFHNN, which is used for the training and inference of HNN and effectively alleviates the common problems of the existing VFL schemes.
- The algorithm can adapt according to the number of participants to flexibly extend the framework to three-party and above scenarios.
- We conduct comparative experiments and demonstrate a reduction of almost 2/3 of required interactions and more than 30% in information transmission to the comparable state-of-the-art scheme, and its performance in all aspects is further improved with the expansion of the framework.

## 2 Related Work

In order to solve the problem of privacy leaks caused by shared information in federated learning [6–10], Gascón et al. [11] proposed a PP protocol for calculating linear regression model based on garbled circuits. Then a series of PP schemes based on partially homomorphic encryption (PHE) and relying on Taylor approximation to deal with non-linear calculations [14,15] appeared one after another. Gu et al. [20] and Zhang et al. [21] proposed VFL schemes supporting SVM with kernels and logistic regression based on random mask and tree-structured communication [28], respectively. Wang et al. [13] proposed the first hybrid DP framework for VFL, which supports GLM, such as logistic regression. Chen et al. [12] solved VFL in an asynchronous fashion based on DP and eliminated peer-to-peer communication required by all participants, which supports logical

regression and neural networks. However, these methods apply only to limited models. They require the third party to be fair and credible and can not collude with other parties. Meanwhile, although these schemes prevent the leakage of shared information, they do not account for potential leakage from the aggregated values themselves [6,8,9,16]. In addition, many schemes with a trusted third party [5,14,22] are not easy to be extended to multi-party scenarios due to the use of PHE. In this paper, we use CKKS FHE scheme [29] that supports arbitrary addition and multiplication of ciphertext to eliminate the limitation of the framework in supporting algorithms and the training and inferential mode of encrypted state to prevent the potential leakage of the aggregated values.

At the same time, a series of frameworks without a third party was proposed. GELU-Net [19] proposes a PP architecture based on PHE to support deep neural networks, which ensures the accuracy and stability of training, but it works under the situation of one party. Zhang et al. [18] proposed a scheme called asymmetrically collaborative machine learning, which focuses on solving the privacy issue that data and labels are distributed on different parties. Based on the above works, FATE launched the existing VFL HNN scheme [17] based on Paillier cryptosystem [23]. This scheme can provide the same accuracy as the methods without privacy protection but introduces many noise-related operations and sharply increases the number of interactions required. In FLFHNN, FHE is used to make the system without additional communication costs and noise-related operations. Aiming at the problem that HE does not support Non-linear calculations in ML, we deal with it by combining the approximation methods proposed by the relevant researches [24–26] for different non-linear calculations and the novel design of separating linear and Non-linear calculations in GELU-Net [19]. At the same time, the scalability of the framework is significantly improved.

## 3 Overview of FLFHNN

This research adopts the semi-honest model, a standard and widely used adversary model. Participants honestly follow the protocol but exploit any opportunity to extract private data from intermediate results generated during the execution of the cryptographic protocol, which is prevented by our work.

For convenience, we simplify the participants into two parties for analysis. The architecture of FLFHNN is shown in Fig. 2, including four main entities:

- **Guest:** Guest provides sample data and labels and receives feature calculation results from Host.
- **Host:** Host only provides sample data without labels and needs to send its own relevant feature calculation results to Guest.
- **Cryptographic service provider (CSP):** It is responsible for distributing public key required for encryption and providing decryption for training and inference results. It also sends decrypted results of inference to Inquirer.
- **Inquirer:** Inquirer sends encrypted inference data to all participants through CSP and obtains final inference results from CSP.

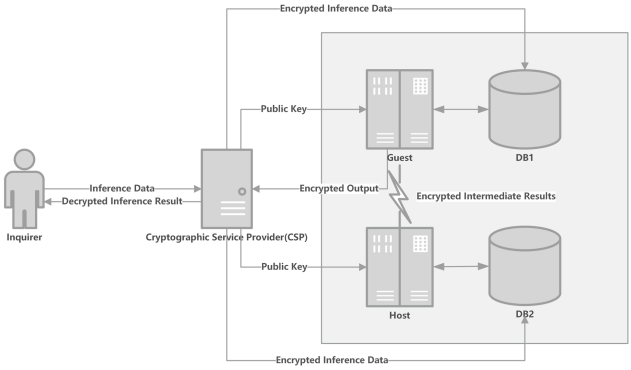



**Fig. 2.** FLFHNN architecture in the two-party scenario

Compared with Paillier based scheme [17], the HNN in FLFHNN has no change in structure and distribution. This model is composed of three layers of fully connected NN. Both parties jointly build the interaction layer, and the model of the interaction layer is only owned by Guest. Each layer has an activation function, and the result is classified by sigmoid or softmax functions. In training tasks, firstly, CSP provides each participant with the public key required for encryption. The data samples are aligned under the encrypted scheme to filter out the training samples with common users or sample IDs. Secondly, both parties execute forward propagation based on their features and models. Next, Host encrypts its relevant feature calculation results and sends them to Guest to complete the whole forward propagation. Finally, Guest obtains the result of the joint training and sends the loss to CSP for decryption to decide whether to continue the training. In predicted tasks, Inquirer first encrypts the inference data through CSP, and then CSP distributes the encrypted data to all participants. Next, all participants execute the joint predicted task. Finally, Guest decrypts the joint inference result through CSP and sends it back to Inquirer.

## 4 Privacy-Preserving Learning Algorithms

In this section, we demonstrate the training and inference process of FLFHNN in the two-party scenario (Fig. 3) and elaborate on the proposed PP forward propagation (Algorithm 1) and backpropagation (Algorithm 2).

Before the training, CSP generates the key pair and sends the public key to all participants. The system selects the corresponding operation mode according to the number of participants. Firstly, all participants execute the forward propagation based on their bottom model and obtain the forward output $\alpha_A, \alpha_B, \cdots, \alpha_N$ of the bottom model (Algorithm 1, line 2–3). Secondly, all participants take the output of the bottom model as the input of the interaction layer. Host encrypt and send the input $[\alpha_A], [\alpha_C], \cdots, [\alpha_N]$ to Guest to calculate the aggregation $[Z]$ of the weight polynomial of the interaction layer (Algorithm 1, line 4–5).

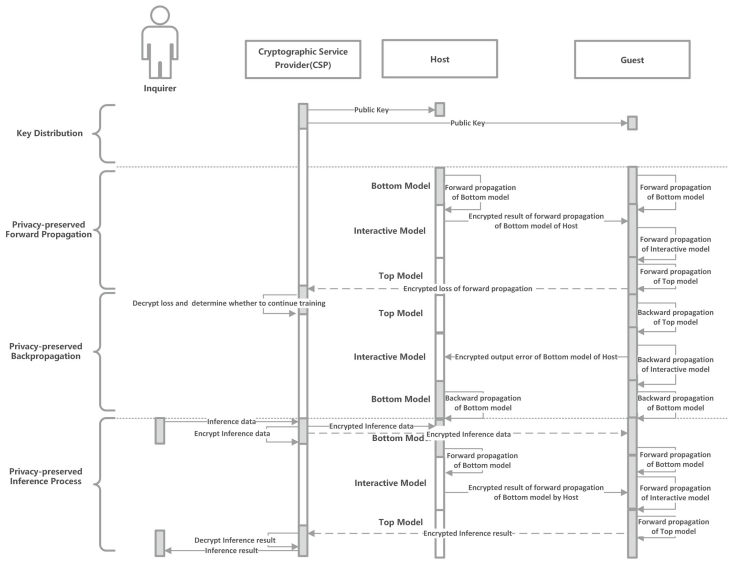



**Fig. 3.** Training and inference process of FLFHNN in the two-party scenario

Next, in the two-party scenario, $[Z]$ is directly input into the activation function to obtain the output $[g(Z)]$ of this layer (Algorithm 1, line 6–7). Finally, Guest feeds $[g(Z)]$ to the top model and executes the forward propagation of this layer (Algorithm 1, line 8). In three-party and above scenarios, Guest first sends the aggregated ciphertext $[Z]$ of the interaction layer to CSP for decryption (Algorithm 1, line 10–11) and then obtains the decrypted information $Z$ to execute the non-linear calculation in plaintext but in a PP manner and gets $g(Z)$ (Algorithm 1, line 12). Finally, Guest feeds $g(Z)$ to the top model and executes the forward propagation of this layer (Algorithm 1, line 13).

The backpropagation is initiated by Guest. In the two-party scenario, first, Guest calculates the loss $[l_{loss}]$ according to the forward propagation result $[p_i]$ and sends it to CSP for decryption $l_{loss} \leftarrow decrypt([l_{loss}])$ to decide whether to continue the training (Algorithm 2, line 2–3). If the condition for continuing training is met, Guest calculates the error $[l_{interactive}] \leftarrow [l_{loss}] \otimes W_{top}$ of the interaction layer and updates the top model (Algorithm 2, line 4–5). Next, Guest calculates the error of the bottom model of itself and Host $[l_{bottom\_B}], [l_{bottom\_A}]$ respectively, and sends $[l_{bottom\_A}]$ to Host and updates the interaction layer model of itself and Host at the same time (Algorithm 2, line 6). Finally, each participant updates its bottom model (Algorithm 2, line 7–8). In three-party and above scenarios, Guest can directly calculate the loss $l_{loss}$ according to the result of the forward propagation (Algorithm 2, step 13) and continue to execute the backpropagation (Algorithm 2, line 14–18). Significantly, some parameters will gradually exist in the form of ciphertext with the increase of training rounds.

**Algorithm 1:** Privacy-preserved Forward Propagation

**Input**: learning rate $\eta$, initialized model $W_i$, training bound $d_{max}$, data samples $x_A, x_B, x_C, \cdots, x_N$

**Output**: output of prediction $[p_i]$ or $p_i$

**for** $d = 1, 2, 3, \cdots, d_{max}$ **do**
  Host: $\alpha_A \leftarrow BottomModel.forwardPropagation(x_A), \alpha_C \leftarrow BottomModel.forwardPropagation(x_C), \cdots, \alpha_N \leftarrow BottomModel.forwardPropagation(x_N)$;
  Guest: $\alpha_B \leftarrow BottomModel.forwardPropagation(x_B)$;
  Host: $[\alpha_A] \leftarrow encrypt(\alpha_A), [\alpha_C] \leftarrow encrypt(\alpha_C), \cdots, [\alpha_N] \leftarrow encrypt(\alpha_N)$. Send $[\alpha_A], [\alpha_C], \cdots, [\alpha_N]$ to Guest;
  Guest: $Z_B \leftarrow W_{interactive\_B}\alpha_B, [Z_B] \leftarrow encrypt(Z_B), [Z_A] \leftarrow W_{interactive\_A} \otimes [\alpha_A], \cdots, [Z_N] \leftarrow W_{interactive\_N} \otimes [\alpha_N], [Z] \leftarrow [Z_A] \oplus [Z_B] \oplus \cdots \oplus [Z_N]$;
  **if** $NumberOfParticipantsi = 2$ **then**
    Guest: Send $[Z]$ to the activation function and get $[g(Z)]$;
    Guest: $[p_i] \leftarrow TopModel.forwardPropagation([g(Z)])$;
  **else**
    Guest: Send $[Z]$ to CSP for decryption;
    CSP: Decrypt $[Z]$, and send $Z$ to Guest;
    Guest: Send $Z$ to the activation function and get $g(Z)$;
    Guest: $p_i \leftarrow TopModel.forwardPropagation(g(Z))$;
  **end**
  Call Algorithm 2 for backpropagation
**end**

## 5 Experiments

In order to verify the effectiveness of the proposed scheme, we implement FLFHNN using TenSEAL library [30] which includes BFV [27] and CKKS [29] implementations, and evaluate FLFHNN based on real-world and publicly available datasets. Specifically, we evaluate and compare the training model accuracy of the scheme without a third party and privacy protection, the previous scheme based on Paillier [17] and FLFHNN, as well as the communication and computational cost of the two encryption schemes of the training phase.

### 5.1 Datasets

We use the following real-world and publicly available datasets: (a) Breast Cancer Wisconsin (Diagnostic) Dataset [31] with $n = 569, d = 30, hl = 2$; (b) Statlog (Vehicle Silhouettes) Dataset [31] with $n = 946, d = 18, hl = 4$. Both datasets are used to execute classification tasks, where $n$ represents the number of samples, $d$ represents the number of input features, and $hl$ represents the number of neurons output by the last layer of NN. The size of data distribution from the two-party scenario to the five-party scenario for each device is shown in Table 1.

**Table 1.** Data distribution from the two-party scenario to the five-party scenario

| Node | N1 (Host) | N2 (Guest) | N3 (Host) | N4 (Host) | N5 (Host) |
|---|---|---|---|---|---|
| Dataset(a) | 109K/56K/45K/35K | 57K/57K/46K/36K | 0K/70K/53K/47K | 0K/0K/53K/47K | 0K/0K/0K/47K |
| Dataset(b) | 72K/52K/44K/36K | 75K/54K/45K/37K | 0K/43K/35K/31K | 0K/0K/34K/29K | 0K/0K/0K/34K |

### 5.2 Experimental Setup

Our experiments are performed on 6 Linux servers with Intel Xeon E5-2620 v3 CPUs running at 2.40 GHz with 24 threads on 12 cores and 2.0 TB memory. We use the Tensorflow 2.2.4 framework and Keras to build our NN baseline,

**Algorithm 2:** Privacy-preserved Backpropagation

**Input**: output of prediction $[p_i]$ or $p_i$ on Guest, Target $t_i$

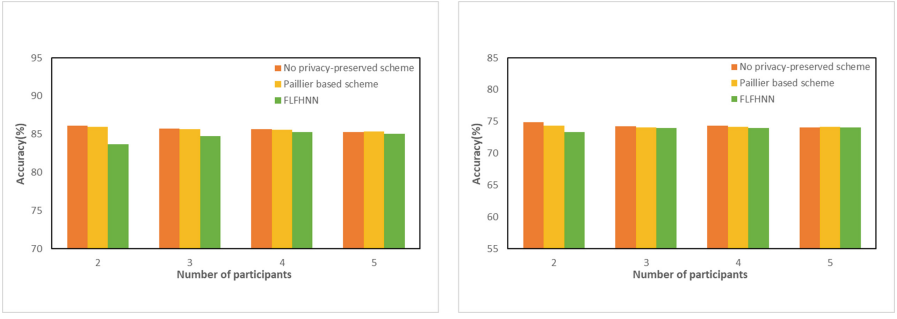
**if** *Number Of Participants* $i = 2$ **then**
  Guest: $[l_{loss}] \leftarrow [t_i] \ominus [p_i]$, and send it to CSP;
  CSP: $l_{loss} \leftarrow decrypt([l_{loss}])$, then compare $l_{loss}$ and the threshold value;
  **if** $l_{loss}$ >*the threshold value* **then**
    Guest: $[l_{interactive}] \leftarrow [l_{loss}] \otimes W_{top}$ and update the top model;
    Guest: $[l_{bottom_B}] \leftarrow [l_{interactive}] \otimes W_{interactive\_B}, [l_{bottom_A}] \leftarrow [l_{interactive}] \otimes W_{interactive\_A}$ and send $[l_{bottom\_A}]$ to Host. Update the interactive models $[W_{interactive\_B\_new}] \leftarrow W_{interactive\_B} \ominus \eta\ \alpha_B \otimes [l_{interactive}], [W_{interactive\_A\_new}] \leftarrow W_{interactive\_A} \ominus \eta\ \otimes [\alpha_A][l_{interactive}]$;
    Host: Updates its bottom model with $[l_{bottom\_A}]$;
    Guest: Updates its bottom model with $[l_{bottom\_B}]$;
  **else**
    End of training;
  **end**
**else**
  Guest: $l_{loss} \leftarrow t_i - p_i$, then compare $l_{loss}$ and the threshold value.
  **if** $l_{loss}$ >*the threshold value* **then**
    Guest: $l_{interactive} \leftarrow l_{loss} W_{top}$ and update the top model;
    Guest: $l_{bottom\_B} \leftarrow l_{interactive} W_{interactive\_B}, l_{bottom\_A} \leftarrow l_{interactive} W_{interactive\_A}, \cdots, l_{bottom\_N} \leftarrow l_{interactive} W_{interactive\_N}$ and send $l_{bottom\_A}, l_{bottom\_C}, \cdots, l_{bottom\_N}$ to Host parties respectively. Update the interactive models $W_{interactive\_B\_new} \leftarrow W_{interactive\_B} - \eta\ \alpha_B l_{interactive}, [W_{interactive\_A\_new}] \leftarrow W_{interactive\_A} \ominus \eta\ l_{interactive} \otimes [\alpha_A], \cdots, [W_{interactive\_N\_new}] \leftarrow W_{interactive\_N} \ominus \eta\ l_{interactive} \otimes [\alpha_N]$;
    Host: Update the bottom models with $l_{bottom\_A}, l_{bottom\_C} \cdots l_{bottom\_N}$ respectively;
    Guest: Updates the bottom model with $l_{bottom\_B}$;
  **else**
    End of training;
  **end**
**end**
Call Algorithm 1 for the next iteration

and our model was trained with SGD optimizer and Adam optimizer for binary classification and multi-class classification tasks of FL at a learning rate of 0.15. For each scheme, we executed five trials and took the average value for statistics.

### 5.3 Experimental Results

Firstly, we evaluated and compared the training model accuracy of the scheme without a third party and privacy protection, the previous scheme based on Paillier [17] and FLFHNN. Secondly, the total computational and communication cost of the two encryption schemes. We further evaluated the computational cost of the two encryption schemes in encryption, decryption, and joint training.

**Accuracy.** For the two-party scenario, the accuracies of the training models of the three schemes (1. No privacy-preserved scheme, 2. Paillier based scheme, 3. FLFHNN) are dataset(a): (86.09%, 85.94%, 83.72%), dataset(b): (74.86%, 74.35%, 73.32%) respectively. The accuracy of FLFHNN is slightly lower than that provided by Paillier based scheme. For three-party and above scenarios, the accuracy of FLFHNN is closer to that provided by Paillier based scheme. Because FLFHNN uses relatively more approximate calculations to deal with non-linear calculations in the two-party scenario, the accuracy of the results is inevitably affected. In three-party and above scenarios, many approximate calculations are eliminated by decryption and return, improving the accuracy (Fig. 4).

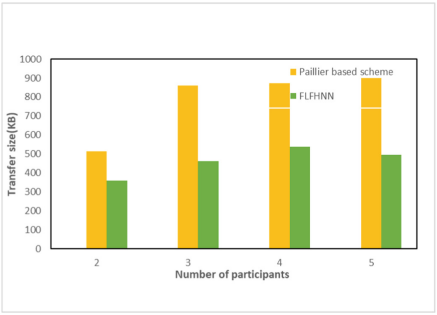



(a) Accuracy of three FL schemes for dataset(a) (b) Accuracy of three FL schemes for dataset(b)

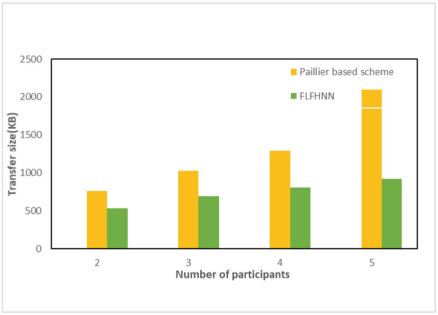
**Fig. 4.** Classification accuracy comparison with different datasets

**Communication Cost.** Compared with Paillier based scheme, the relationships of FLFHNN in terms of the size of information transmission of different number of participants (1. two parties, 2. three parties, 3. four parties, 4. five parties) are dataset(a): (−30.12%, −46.04%, −38.18%, −44.71%), dataset(b): (−30.38%, −32.80%, −37.80%, −56.23%) respectively. Significantly, compared with Paillier based scheme, FLFHNN needs two additional interactions between Guest and CSP in each round of training. However, the overall information transmission

required by FLFHNN is still much smaller. Furthermore, this advantage becomes more and more obvious with the increase in the number of participants. And this is also an inevitable result of frequent interactions between the participants and a large number of noise-related operations of Paillier based scheme.



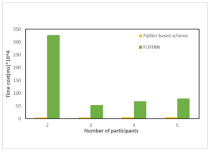

(a) Communication cost of two encryption FL schemes for dataset(a)



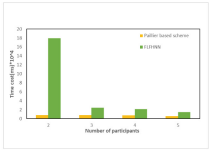

(b) Communication cost of two encryption FL schemes for dataset(b)

**Fig. 5.** Communication cost comparison with different datasets in a round of training

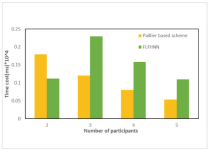
**Computational Cost.** In the two-party scenario, compared with Paillier based scheme, the relationships of FLFHNN in terms of total computational cost, encryption cost, decryption cost and computational cost of joint training are dataset(a): (+8286.27%, +2148.19%, −37.68%, +10479.29%), dataset(b): (+1412.55%, +3555.08%, −23.45%, +818.67%) respectively. In the three-party



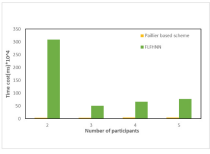
(a) Computational cost for dataset(a)



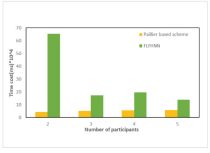
(b) Encryption cost for dataset(a)



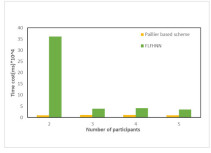
(c) Decryption cost for dataset(a)



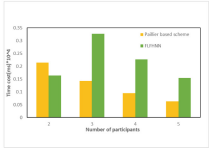
(d) Joint training cost for dataset(a)



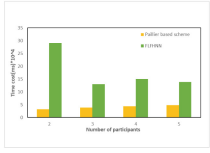
(e) Computational cost for dataset(b)



(f) Encryption cost for dataset(b)



(g) Decryption cost for dataset(b)



(h) Joint training cost for dataset(b)

**Fig. 6.** Computational cost of two encryption FL schemes for different datasets in a round of training

scenario, dataset(a): (+1093.66%, +198.46%, +97.37%, +1340.09%), dataset(b): (+235.89%, +249.66%, +129.78%, +234.68%). In the two-party scenario, FLFHNN adopts the training mode of encrypted state, and its total computational cost is higher than Paillier based scheme. However, with the increase of participants, this trend gradually slows down, mainly because the multi-party algorithm of FLFHNN significantly reduces the calculations between ciphertexts (Figs. 5 and 6).

**Summary.** From these tests we find the following.

1. Although the training accuracy of FLFHNN is slightly lower than that of Paillier based scheme in the two-party scenario, it basically ensures the availability of the model. In multi-party scenarios, the accuracy of FLFHNN is almost the same as that of the previous scheme.
2. FLFHNN has obvious advantages over the previous scheme in communication cost. The sizes of information transmission are reduced by dataset(a): (−30.12%) and dataset(b): (−30.38%) in the two-party scenario. Furthermore, this advantage is more and more obvious with the increase of the participants.
3. Compared with the previous scheme, FLFHNN reduces the computational complexity by eliminating noise-related operations, but its overall computational cost is still higher. However, this problem has been significantly improved by the algorithm adapting to the number of participants. In the future, GPU can also be used to accelerate computational efficiency between the ciphertexts of FHE.

## 6 Conclusion and Future Work

We propose FLFHNN, an efficient and flexible VFL HNN framework based on CKKS FHE scheme, which effectively alleviates the common problems of PP VFL. The experimental results show that compared with the previous scheme, FLFHNN improves the interactive efficiency between participants on the premise of ensuring the model accuracy and has robust scalability. It is applicable in the internet of things scenarios with many devices and large-scale distributed data. Future research will focus on VFL building security defense systems against malicious attacks such as collusion, poisoning, and adversarial attacks.

**Acknowledgments.** This work is supported by the Cooperation project between Chongqing Municipal undergraduate universities and institutes affiliated to CAS (HZ2021015).

## References

1. McMahan, H.B., et al.: Federated learning of deep networks using model averaging. CoRR abs/1602.05629 (2016)

2. Konečný, J., et al.: Federated learning: strategies for improving communication efficiency. CoRR abs/1610.05492 (2016)
3. McMahan, B., et al.: Communication-efficient learning of deep networks from decentralized data. In: AISTATS, pp. 1273–1282. PMLR (2017)
4. Ramaswamy, S., et al.: Federated learning for emoji prediction in a mobile keyboard. CoRR abs/1906.04329(5) (2019)
5. Yang, Q., et al.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. **10**(2), 12:1–12:19 (2019)
6. Zhu, L., Han, S.: Deep leakage from gradients. In: Yang, Q., Fan, L., Yu, H. (eds.) Federated Learning. LNCS (LNAI), vol. 12500, pp. 17–31. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63076-8_2
7. Phong, L.T., et al.: Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans. Inf. Forensics Secur. **13**(5), 1333–1345 (2018)
8. Melis, L., et al.: Exploiting unintended feature leakage in collaborative learning. In: SP, pp. 691–706. IEEE (2019)
9. Hitaj, B., et al.: Deep models under the GAN: information leakage from collaborative deep learning. In: CCS, pp. 603–618. ACM (2017)
10. Wang, Z., et al.: Beyond inferring class representatives: user-level privacy leakage from federated learning. In: INFOCOM, pp. 2512–2520. IEEE (2019)
11. Gascón, A., et al.: Secure linear regression on vertically partitioned datasets. IACR Cryptology ePrint Archive 892 (2016)
12. Chen, T., et al.: VAFL: a method of vertical asynchronous federated learning. CoRR abs/2007.06081 (2020)
13. Wang, C., et al.: Hybrid differentially private federated learning on vertically partitioned data. CoRR abs/2009.02763 (2020)
14. Hardy, S., et al.: Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. CoRR abs/1711.10677 (2017)
15. Yang, K., et al.: A quasi-newton method based vertical federated learning framework for logistic regression. CoRR abs/1912.00513 (2019)
16. Nasr, M., et al.: Comprehensive privacy analysis of deep learning: passive and active white-box inference attacks against centralized and federated learning. In: SP, pp. 739–753. IEEE (2019)
17. WeBank AI Department. https://github.com/FederatedAI/FATE. Accessed 9 May 2022
18. Zhang, Y., Zhu, H.: Additively homomorphical encryption based deep neural network for asymmetrically collaborative machine learning. CoRR abs/2007.06849 (2020)
19. Zhang, Q., et al.: GELU-Net: a globally encrypted, locally unencrypted deep neural network for privacy-preserved learning. In: IJCAI, pp. 3933–3939. ijcai.org (2018)
20. Gu, B., et al.: Federated doubly stochastic kernel learning for vertically partitioned data. In: KDD, pp. 2483–2493. ACM (2020)
21. Zhang, Q., et al.: Secure bilevel asynchronous vertical federated learning with backward updating. In: AAAI, pp. 10896–10904. AAAI Press (2021)
22. Kim, M., et al.: Secure logistic regression based on homomorphic encryption. IACR Cryptology ePrint Archive 74 (2018)
23. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44586-2_9
24. Gilad-Bachrach, R., et al.: CryptoNets: applying neural networks to encrypted data with high throughput and accuracy. In: ICML, pp. 201–210. JMLR.org (2016)

25. Hesamifard, E., et al.: CryptoDL: deep neural networks over encrypted data. CoRR abs/1711.05189 (2017)
26. Chou, E., et al.: Faster CryptoNets: leveraging sparsity for real-world encrypted inference. CoRR abs/1811.09953 (2018)
27. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. IACR Cryptology ePrint Archive 144 (2012)
28. Zhang, G.-D., et al.: Feature-distributed SVRG for high-dimensional linear classification. CoRR abs/1802.03604 (2018)
29. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 409–437. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_15
30. OpenMined. https://github.com/OpenMined/TenSEAL. Accessed 9 May 2022
31. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets.php. Accessed 9 May 2022

# Phishing Frauds Detection Based on Graph Neural Network on Ethereum

Xincheng Duan[1], Biwei Yan[1,2,3], Anming Dong[1,2,3], Li Zhang[1,2,3(✉)], and Jiguo Yu[2,3]

[1] School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, Shandong , People's Republic of China
xinchengduan@yeah.net, {anmingdong,lizhang}@qlu.edu.cn

[2] Big Data Institute, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, Shandong, People's Republic of China
jiguoyu@sina.com

[3] Shandong Fundamental Research Center for Computer Science, Jinan 250300, People's Republic of China

**Abstract.** Blockchain, as an emerging technology, has vulnerabilities that make the blockchain ecosystem rife with many criminal activities. However, existing technologies of phishing fraud detection heavily rely on shallow machine learning, leading to low detection precision. To solve this problem, in this paper, we construct a graph classification network model TransDetectionNet. Particularly, we propose a node embedding algorithm named *Edge-sampling To Node Vector* (Esmp2NVec) that can effectively extract the features hiding in the directed transaction network. Then, we use graph convolutional neural networks (GraphSAGE and GCN) to learn the topological space structure between nodes for further extraction of node features, where the nodes represent Ethereum accounts. To evaluate the method, a series of transaction data from the real Ethereum system is leveraged to train the graph classification model, and several experiments are designed to measure the phishing accounts identification performance comparison between our method and the other related works. The final results of those experiments show that our method can effectively detect phishing accounts from the Ethereum system.

**Keywords:** Blockchain · Ethereum · Phishing detection · Graph classification

This work was supported in part by the NSF of China under Grants 61832012, 61771289 and 61902202, and the Piloting Fundamental Research Program for the Integration of Scientific Research, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant 2022XD001.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 351–363, 2022.
https://doi.org/10.1007/978-3-031-19208-1_29

## 1 Introduction

Blockchain, a decentralized distributed ledger, is the underlying technology to realize the point-to-point transfer of digital assets [1]. The rapid development of blockchain technology has attracted the extensive attention of researchers in various fields and has had a far-reaching impact on the fields of finance, science, and technology. Ethereum is a decentralized smart contract platform based on blockchain technology [2]. The Ethereum platform allows anyone to build applications based on blockchain technology.

However, with the rapid development of the Ethereum cryptocurrency market as well as the imperfection of the Ethereum ecosystem, there are many inevitable criminal activities of online fraud, such as Ponzi schemes, phishing fraud, and so on [3]. The latest report from the Kaspersky Lab shows that Ethereum is currently the favorite cryptocurrency of phishers. According to a report by Chainalysis, more than half of all cybercrime revenue in Ethereum since 2017[1] is related with phishing frauds. A typical example is the phishing fraud on BeeToken ICO[2]. The phishing fraud ultimately defrauded investors of approximately $1 million in just 25 h.

To effectively extract the characteristics of phishing accounts, identify potential phishing accounts, and develop a more healthy and secure blockchain ecosystem, we model transaction-based blockchain phishing detection as a graph classification task. In this paper, we construct a transaction network graph for each target account from the account's transaction records and design the Edge-sampling To Node Vector (Esmp2NVec) algorithm to generate feature vectors for each vertex in the transaction network graph. Furthermore, we use the GraphSAGE and GCN to deeply extract the spatial structure and feature information of vertices in the transaction network graph and construct a graph classification model for the classification of transaction network graphs. Where each transaction network graph represents a target account.

The main contributions of this paper are summarized as follows:

1) A simple transaction pattern graph is the first to define for each account to be tested. Specifically, we generate a simple directed transaction pattern graph centered on itself for each node to be tested, to reduce the complexity of the data structure.
2) We propose the Edge-sampling To Node Vector (Esmp2NVec) algorithm to generate the features of the vertices. The algorithm samples from the edges that connected vertex and generates a vertex embedding for each vertex that contains the flow direction of funds.
3) To extract transaction information and topological space structure features in the transaction network, we designed the graph classification model TransDetectionNet for extracting deeper feature information in the transaction pattern graph and obtaining higher quality graph embeddings for graph recognition and classification.

[1] https://blog.chainalysis.com/the-rise-of-cybercrime-on-ethereum/.
[2] https://theripplecryptocurrency.com/bee-token-scam/.

## 2 Related Work

At present, many machine learning, deep learning and other methods have been widely used in Image, NLP, Fraud Detection and other fields [4,5]. Although the existing anomaly detection methods perform well in the task of detecting phishing in Ethereum, these methods still have limitations. The problem with the method of recognizing phishing accounts through a shallow learning mechanism is that it relies too much on feature engineering [6]. The structural graph is making a splash in fields such as social, so existing work considers abstracting node transaction data into a transaction graph and transforming the identification of phishing nodes into a node classification task [7]. When extracting the features of the nodes in the transaction network, several researchers have used the walk-based network embedding method to generate the feature vectors of the nodes [8]. However, the walk-based network embedding method fails to take into account the edge weight information when extracting features and cannot do deep feature extraction in the transaction network.

Existing phishing fraud detection methods mainly extract network features through graph embedding algorithms random walk-based. Wu, et al. proposed a method, which is called Trans2Vec, to detect Ethereum phishing fraud by mining Ethereum transaction records [8]. However, it did not consider the direction of the transaction. Chen, et al. proposed a graph-based cascade feature extraction method based on transaction records and a lightGBM-based Dual-sampling Ensemble algorithm to build the identification model [9]. Wen, et al. proposed a phishing detection framework based on feature learning and a phishing concealment framework based on inserting transaction records, which enhanced the robustness of the phishing framework and further improved the phishing accounts recognition rate [10]. Some researchers have also considered the Graph Neural Network for blockchain fraud detection [11].

## 3 Preliminary

This section introduces the definition of the transaction graph and constructs a transaction pattern graph using real transaction records of Ethereum. To easily describe the transactions between nodes and extract the transaction features, we use the weighted directed graph to describe the transaction network between nodes.

A transaction network centered on a node to be tested can be expressed as $G = (V, E, F, L)$, where $V$ represents the node-set and $E$ is the edge-set. Edge attribute $F \in \mathbb{R}^{|E|\times|\boldsymbol{a}|}$, $\boldsymbol{a}$ is the feature vector of an edge, and $|\boldsymbol{a}|$ is the size of the set. There may be multiple transactions between two nodes. We take the average value of multiple transactions in the same direction as one of the features of the simple graph edge:

$$\boldsymbol{a_0} = \frac{\sum_{v_i, v_j \in V}^{n} \boldsymbol{M}_{<v_i, v_j>}}{\boldsymbol{n}} \tag{1}$$

where $\boldsymbol{a_0}$ is the first feature value of $\boldsymbol{a}$, $\boldsymbol{M}_{<v_i,v_j>}$ represents the each transfer amount from node $v_i$ to node $v_j$, and n represents the number of transfers. The node attribute $L \in \mathbb{R}^{|V|\times|l|}$, where $l$ is the label set of the node. In addition, we extract the data related to the maximum value, minimum value, and the number of transactions of multiple transaction data existing in the same direction for two transaction addresses over a period of time as edge feature information.

To facilitate data processing, we number the addresses to be tested and the addresses that have transactions with the addresses to be tested. As shown in Fig. 1(a), we replace the account addresses with numbers and divide the nodes in the transaction into two categories: target accounts and other accounts. We label the target account as type A, and the type of other accounts is labeled as B. Then, we construct the transaction between the target account and other accounts into a weighted directed graph centered on the target account.

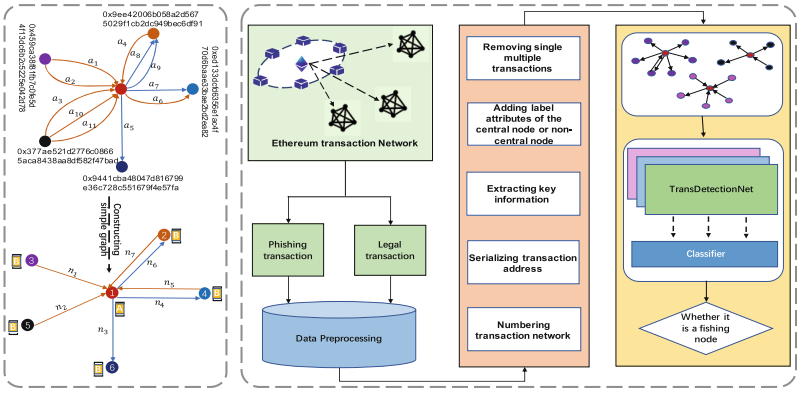



**Fig. 1.** (a) Directed graph describes transactions network. (b) The overall framework of Ethereum phishing detection

## 4 Our Method

The overall architecture of our method is shown in Fig. 1(b), which firstly obtains transaction records of accounts from Ethereum through an automated program; secondly, analyzes and processes the transaction data and constructs them into trainable transaction graphs; then, classifies the transaction graphs by our proposed model. This section will specifically introduce the different parts of the graph classification model proposed in this paper.

### 4.1 Edge-Sampling to Node Vector Algorithm

Based on the transaction pattern graph constructed above, we propose the Edge-sampling To Node Vector (Esmp2NVec) algorithm. That generates the initial

feature vector of the corresponding vertex by sampling from the edges connected to the vertex.

**Algorithm 1:** Esmp2NVec algorithm on the transaction network graph

**Input**: The transaction network G=($V, E, F, L$), where $L$ contains category labels for each vertice in the graph, $F$ contains the transaction amount information of all edges, embedding dimension $d$.
**Output**: The feature matrix of vertices $X(|V|, d)$
i 0;
**while** i < $|V|$ **do**
  in-degree matrix filled with zeros: $I_v$; out-degree matrix filled with zeros: $O_v$.
  j 0, $n_I$ 0, $n_O$ 0;
  **while** j < ($|E[0]|$ or $|E[1]|$) and $n_O < (d/2 - 1)$ and $n_I < (d/2)$ **do**
    *value* $F$[j][0];
    **if** $E$[0][j] == i **then**
      $f$ (-1) * *value*;
      $O_v[n_O]$ $f$;
      $n_O$ $n_O + 1$
    **if** $E$ [1][j] == i **then**
      $f$ *value*;
      $I_v[n_I]$ $f$;
      $n_I$ $n_I + 1$;
    j j + 1;
  $X$[i] features of the merger ($L$[i], $I_v$, $O_v$);
return $X$;

Algorithm 1 describes the extraction process of vertex features in a single directed transaction network. Each step of the outer loop of Algorithm 1 performs sequentially: where i represents the number of the vertices in the currently directed graph, and $|V|$ represents the number of vertices in the currently directed graph. Both the $I_v$ and the $O_v$ are a one-dimensional matrix filled with zero, which is used to store the feature values on the in and out edges connected to the vertex. If there is a value on the edge connected to the vertex, the value on the edge is processed by Formula (2) and then filled into the corresponding position of the $I_v$ or $O_v$ matrix, and the other positions default to zero. The inner loop is to traverse the edges in the graph, where j represents the index of the edge, $|E[0]|$ or $|E[1]|$ represents the total number of edges; $n_I$ and $n_O$ are used to count the number of in-degree and out-degree of the current vertex, and used to determine the position of the weight value in the $I_v$ or $O_v$ matrix. After the inner loop is finished, the obtained $I_v$ and $O_v$ matrices are combined with the label matrix $L$[i] to obtain the features describing the current vertex. After the weighted directed transaction network passes the Esmp2NVec algorithm, we will finally get the Network embedding as shown in Fig. 2.

$$f = \begin{cases} value, & \text{(if the edge is an in-degree edge)} \\ (-1) * value, & \text{(if the edge is an out-degree edge)} \end{cases} \tag{2}$$

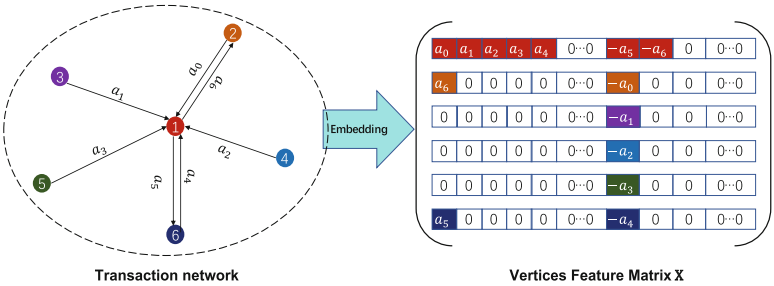



**Fig. 2.** Generating node embedding of the transaction network

## 4.2 Model Construction

In this part, we will introduce the our proposed network model TransDetectionNet for Ethereum phishing detection. Figure 3 shows the model diagram of phishing detection. Our model mainly contains three modules, the Esmp2NVec module, the inductive representation learning module, and the graph convolutional neural network layer.

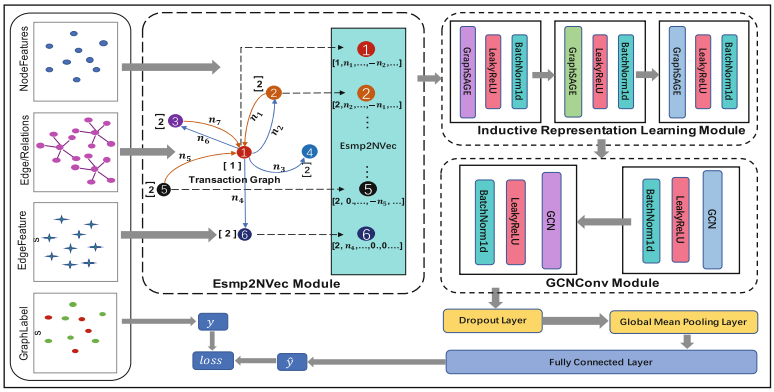



**Fig. 3.** The TransDetectionNet model

Esmp2NVec Module: It mainly extracts the transaction information existing on the edges of the transaction network, and generates a vector representation for the vertices of the transaction graph. For example, the initial feature vector of node $N_v$, $\forall v \in V$ is only composed of labels. After algorithm 1, the vector of node $N_v$ is expressed as $n_v = [1, a_1, a_2, ..., a_k, 0, ..., a_{(k+1)}, a_{(k+2)}, ..., 0, ...]$ where $a_k$ is the value calculated by formula (2). Finally, we get the feature matrix $X$ of the node, as shown in Fig. 2.

Inductive Representation Learning Module: GraphSAGE is an inductive algorithm, which learns feature information from vertex neighborhoods by training a set of aggregation functions. The running process of GraphSAGE is roughly divided into three steps: (1) Random sampling of neighbors, where the number of neighbors sampled in each hop is not more than $S_k$. (2) Aggregating the information contained in neighbor vertices through the aggregation function. (3) Generating a vector representation of the target node for downstream tasks. In this paper, we choose a Mean aggregator to aggregate the information contained in neighbor vertices. The specific aggregate function formula is as follows:

$$h_v^k = \sigma\left(W \cdot MEAN(\{h_v^{k-1}\} \cup \{h_u^{k-1}, \forall u \in \mathcal{N}(v)\})\right) \tag{3}$$

The feature vector of the nodes obtained in module 1 is $n_v \in X, \forall v \in V$. This feature vector will be used as the initial feature vector input of GraphSAGE: $h_v^0 \leftarrow n_v, \forall v \in \mathrm{V}$, after K iterations, each node will get a higher quality vector representation: $n_v \leftarrow h_v^K, \forall v \in \mathrm{V}$, thus, a new feature matrix $X$ is obtained as the initial feature input of the next module.

Graph Convolutional Neural Network (GCN) Module: GCN is an effective variant of a convolutional neural network based on graph operations, which can effectively extract nodes' feature information and topological spatial structure in the graph. Multi-layer graph convolutional network (GCN), its inter-layer propagation rules are as follows:

$$H^{(l+1)} = f(H^l, A) = \sigma\left(\hat{A}\widetilde{H}^{(l)}W^{(l)}\right) \tag{4}$$

The GCN implementation process is: (1) The $\hat{A}\widetilde{H}^{(l)}$ operation is a feature transfer between nodes, for aggregating information from surrounding nodes to update the current node. (2) The $\sigma\left(\hat{A}\widetilde{H}^{(l)}W^{(l)}\right)$ is to perform a linear transformation on each node and use the activation function to activate it. (3) Repeating steps (1) and (2) L times to achieve multi-layer convolution. (4) Obtaining the final $H^L$ as the matrix representation of the node. In this paper, we construct a two-layer GCN, with LeakyReLU as the activation function. Finally, we get the node feature calculation formula:

$$X = LeakyReLU\left(\hat{A}LeakyReLU\left(\hat{A}XW^{(0)}\right)W^{(1)}\right) \tag{5}$$

### 4.3 Model Training

In the model training process, to speed up the training and convergence speed of the model, and increase the stability of the model, we use BatchNorm1d to normalize the output data of each layer. We found that overfitting occurred during model training. In response to the overfitting situation, we use global average pooling to average pool the global data before processing the final classification of the data and adding a dropout layer to achieve the purpose of reducing network parameters and preventing overfitting.

Then we send the resulting feature matrix $X$ to the fully connected layer for final classification, and get the prediction result:

$$\hat{y} = softmax(XA^T + b) \tag{6}$$

where $A$ is the parameter matrix and b is the bias.

We use the CrossEntropyLoss function during model training. This function combines two functions of LogSoftmax and NLLLoss, and the calculation formula for the loss of a single sample in the binary classification can be expressed as:

$$loss = -[y \cdot log(\hat{y}) + (1 - y) \cdot log(1 - \hat{y})] \tag{7}$$

where $\hat{y}$ represents the predicted value and $y$ represents the real value.

## 5 Experiments

To verify the effectiveness of our methods, we conducted multiparty experiments on Ethereum's real transaction data. In this section, we first introduce the dataset, the baseline method, the related experimental setup, and the evaluation metrics. Then we give the experimental results as well as analyze the experimental results.

### 5.1 Dataset Description

The dataset used in our experiment was downloaded from the XBlock[3] website. XBlock collects current mainstream blockchain data and is one of the popular blockchain data platforms in academia. This dataset is the real second-order historical transaction data of 3360 source accounts obtained from Ethereum. By analyzing the data, we found that the first-order transaction data on Ethereum is the most representative of the features of the account. Therefore, for reducing the complexity of the data, we take the first-order transaction data of the target account as the dataset for the experiments. We randomly selected phishing nodes and non-phishing nodes in the dataset to the official Ethereum[4] website to verify the validity of the dataset. There are 1,660 nodes marked as phishing nodes in this dataset, and another 1,700 nodes are non-phishing nodes. Therefore, the ratio of positive and negative samples of the constructed transaction graph is close to 1:1. In our experiments, we mix two different types of transaction graph data together for random shuffle and use 70% of them as the training dataset and the rest of them as the validation dataset.

[3] http://xblock.pro/tx-cn-2/.
[4] https://eth.bitaps.com.

### 5.2 Baseline Methods

In this experiment, we use the Esmp2NVec algorithm to obtain the feature vector for describing the node. The method we proposed will be compared with the popular graph node embedding methods, such as DeepWalk and node2vec, in the same experimental environment.

1) DeepWalk [12]: DeepWalk algorithm is similar to word2vec. The algorithm uses the co-occurrence relationship between nodes in the graph to learn the vector representation of nodes.
2) node2vec [13]: node2vec uses the Alias algorithm for vertex sampling, which is an extension of DeepWalk. The algorithm combines BFS and DFS to explore the structure and homogeneity of the graph.
3) Walklets [14]: Walklets have made some improvements to the deficiencies of DeepWalk. It can capture the relationship between nodes with larger spatial scales.
4) GCN [15]: GCN is a scalable method for semi-supervised learning on graph structure data, which can effectively extract spatial features of topological graphs.
5) GATConv [16]: The Attention mechanism has been successfully used in many sequence-based tasks. GATConv can assign different attention scores to each neighbor, to identify more important neighbors.
6) GraphSAGE [17]: Instead of training embeddings individually for each node, GraphSAGE learns a set of functions that generate embeddings by sampling and aggregating features from the local neighborhoods of the nodes.

### 5.3 Experimental Setup and Evaluation Metrics

In our model, we use dropout layers to alleviate overfitting. The probability p that an element is zeroed out is set to 0.5. During model training, Adam is used as the optimization algorithm for gradient descent. L2 regularization is added and the learning rate is dynamically adjusted using the MultiStepLR function, the gamma factor of learning rate decay is set to 0.1. The Batch size is set to 64. We repeated each experiment 5 times independently, and the epoch of each experiment was set to 200, and the average of the 5 times results was taken as the final result. To evaluate the performance of different methods in the phishing detection task fairly and squarely, we considered four evaluation metrics, namely, accuracy, precision, recall, and F1-score.

### 5.4 Performance Analysis

Table 1 shows the experimental results of the Esmp2NVec algorithm and the walk-based graph embedding method. The embedding size of all embedding methods is 128. From the experimental results, the overall performance effect of the walk-based graph node embedding algorithm is relatively poor. This is because the above-mentioned walk-based graph node embedding algorithm,

**Table 1.** The following table shows the performance of SVM classification under different graph embeddings

| Method | *Accuracy* | *Precision* | *Recall* | *F1-score* |
|---|---|---|---|---|
| DeepWalk | 0.705 | 0.701 | 0.719 | 0.706 |
| node2vec | 0.639 | 0.661 | 0.569 | 0.607 |
| Walklets | 0.703 | 0.701 | 0.714 | 0.704 |
| **Esmp2NVec** | **0.924** | **0.889** | **0.965** | **0.924** |

**Table 2.** Performance comparison between TransDectionNet and mainstream graph neural network models

| Method | *Accuracy* | *Precision* | *Recall* | *F1-score* |
|---|---|---|---|---|
| GCNConv | 0.933 | 0.911 | 0.953 | 0.931 |
| GATConv | 0.712 | 0.707 | 0.678 | 0.688 |
| GraphSAGE | 0.645 | 0.620 | 0.665 | 0.637 |
| **TransDectionNet** | **0.946** | **0.976** | **0.913** | **0.943** |

when generating node embedding, mainly simply imitates the relationship between words and words, ignoring the weight of edges. However, the key information of the transaction in the transaction network exists on the edge, which causes the actual performance of the above-mentioned walk-based graph node embedding algorithm in the experiment to be relatively poor. However, our proposed Esmp2Nvec algorithm takes into account the transaction direction in the transaction network when generating the vertex embedding, so the actual performance Esmp2NVec method is much better than the walk class graph embedding method.

Table 2 is the experimental result of the recently popular graph neural network model and TransDetectionNet. The results show that the performance of GraphSAGE and GAT perform poorly in our experiment. Because Both GAT and GraphSAGE are unable to take full advantage of the transaction information on the edge of the trading network, which leads to the poorer actual performance of both in the experiments. GCN is able to utilize one-dimensional features of the edges, the weights of the edges. Since the most effective information in the Ethernet transaction network is located at the edge connecting each node, the GCN method performs best in the practice of the message passing model. To further extract the transaction features in the transaction network, this paper designs the TransDetectionNet graph neural network model to obtain higher-quality transaction network embeddings, which further improves the recognition rate of phishing nodes.

From the experimental results, our model has obtained a good result in extracting features of accounts through transactions to identify potential

phishing nodes. In addition, our method can also be used for the detection of other fraudulent scams, such as the detection of Ponzi schemes, etc.

### 5.5 Model Parameters Study

**Effect of Layer Numbers.** Figure 4(a) is the result of an experiment on SAGEConv of different depths without a GCNConv module. We observe that the precision of TransDetectionNet reaches its maximum when the depth of SAGEConv is 3. Then, the accuracy value decreases slightly with the increase of SAGEConv depth, while the other metrics do not change significantly with the increase of SAGEConv depth. We fixed the depth of SAGEConv to 3, and successively increased the number of layers of GCNConv to obtain the experimental results shown in Fig. 4(b). From the experimental results, we can see that the comprehensive performance of TransDetectionNet is better when GCNConv is two layers. Finally, in the TransDetectionNet proposed in this paper, we set the depth of the SAGEConv layer to 3 and the depth of the GCNConv layer to 2.

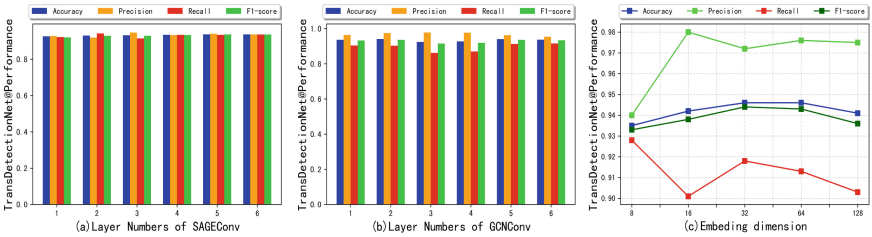



**Fig. 4.** The impact of the number of layers of SAGEConv and GCNConv on the performance of TransDetectionNet

**Effect of Embedding Dimension.** The embedding dimension of the vector also has a certain impact on the classification performance of the model. From the experimental results shown in Fig. 4(c), we observe that when the dimension of the vector is 8, the performance of the model is the worst. The F1-score of the model is relatively better when the vector dimension is 32 or 64. When the vector dimension is 128, each indicator has a different degree of decline, which shows that too large a vector dimension may lead to a certain overfitting phenomenon in the model.

## 6 Conclusion and Future Work

In this paper, we have developed an in-depth study on the detection of phishing accounts on Ethereum. By analyzing the characteristics of phishing fraud transactions and normal transactions in Ethereum, we propose the Esmp2NVec

node embedding algorithm, which generates node embedding containing the flow of funds for each node in the transaction network. Furthermore, to reduce the computational complexity and further improve the precision of phishing network recognition, we design the graph classification model TransDetectionNet. Although our method shows good performance in feature extraction for transaction networks, this fraud detection method relies excessively on account transaction data. In the future, we will further extend our fraud detection methods to deal with the limitations of over-reliance on transaction data.

## References

1. Zheng, Z., Xie, S., Dai, H.-N., Chen, X., Wang, H.: Blockchain challenges and opportunities: a survey. Int. J. Web Grid Serv. **14**(4), 352–375 (2018)
2. Chen, J.: Finding ethereum smart contracts security issues by comparing history versions. In: 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 1382–1384 (2020)
3. Poursafaei, F., Hamad, G.B., Zilic, Z.: Detecting malicious Ethereum entities via application of machine learning classification. In: 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), pp. 120–127. IEEE (2020)
4. Yuan, Q., Huang, B., Zhang, J., Wu, J., Zhang, H., Zhang, X.: Detecting phishing scams on ethereum based on transaction records. In: 2020 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5 (2020)
5. Cai, Z., Xiong, Z., Honghui, X., Wang, P., Li, W., Pan, Y.: Generative adversarial networks: a survey toward private and secure applications. ACM Comput. Surv. (CSUR) **54**(6), 1–38 (2021)
6. Li, Y., Zhenguo Yang, X., Chen, H.Y., Liu, W.: A stacking model using URL and HTML features for phishing webpage detection. Future Gener. Comput. Syst. **94**, 27–39 (2019)
7. Cai, Z., He, Z., Guan, X., Li, Y.: Collective data-sanitization for preventing sensitive information inference attacks in social networks. IEEE Trans. Dependable Secure Comput. **15**(4), 577–590 (2016)
8. Wu, J., et al.: Who are the phishers? Phishing scam detection on ethereum via network embedding. IEEE Trans. Syst. Man Cybern. Syst. **52**(2), 1156–1166 (2022)
9. Chen, W., Guo, X., Chen, Z., Zheng, Z., Lu, Y.: Phishing scam detection on ethereum: towards financial security for blockchain ecosystem. In: IJCAI, pp. 4506–4512 (2020)
10. Wen, H., Fang, J., Wu, J., Zheng, Z.: Transaction-based hidden strategies against general phishing detection framework on ethereum. In: 2021 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5 (2021)
11. Zhuang, Y., Liu, Z., Qian, P., Liu, Q., Wang, X., He, Q.: Smart contract vulnerability detection using graph neural network. In: IJCAI, pp. 3283–3290 (2020)
12. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
13. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)

14. Perozzi, B., Kulkarni, V., Chen, H., Skiena, S.: Don't walk, skip! Online learning of multi-scale network embeddings. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, pp. 258–265 (2017)
15. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. CoRR, abs/1609.02907 (2016)
16. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. arXiv eprint arXiv:1710.10903 (2018)
17. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates Inc. (2017)

# Blockchain-Aided Hierarchical Attribute-Based Encryption for Data Sharing

Jiaxu Ding[1], Biwei Yan[1,2,3], Guijuan Wang[1,2,3](✉), Li Zhang[1,2,3], Yubing Han[1,2,3], Jiguo Yu[2,3], and Yan Yao[1,2,3]

[1] School of Computer Science and Technology, Qilu University of Technology, Jinan 250353, Shandong, People's Republic of China

[2] Big Data Institute, Qilu University of Technology, Jinan 250353, Shandong, People's Republic of China

yaoyan@qlu.edu.cn

[3] Shandong Fundamental Research Center for Computer Science, Jinan 250300, People's Republic of China

**Abstract.** Ciphertext-policy attribute-based encryption (CP-ABE) is widely used in access control to achieve secure data sharing over different cloud platforms. However, most of the existing CP-ABE data sharing schemes perform one-time encryption on the shared data, which cannot satisfy the need for data sharing in the public cloud with complex users. In order to realize multi-user data sharing on blockchain and achieve hierarchical decryption of privacy data and shared data, we propose a blockchain-aided hierarchical and searchable attribute-based encryption scheme, named BC-HSABE. In BC-HSABE, we adopt a symmetric encryption algorithm to encrypt data in a hierarchical manner, and use attribute-based encryption technology to encrypt two hierarchical symmetric keys of the encrypted data file, and upload the ciphertext to the cloud server. In addition, data users can decrypt data ciphertext at different levels according to their hierarchical authority. Meanwhile, searching for keyword trapdoor through blockchain ensures the security of keyword ciphertext. The security analysis and experimental evaluations verify the feasibility and effectiveness of BC-HSABE.

**Keywords:** Blockchain · Data sharing · Hierarchical encryption · Cloud · Fine-grained access control

This work was supported in part by the NSF of China under Grants 61832012 and 61771289, in part by the NSF of Shandong Province under Grant ZR2021QF079 and the Pilot Project for Integrated Innovation of Science, and the Piloting Fundamental Research Program for the Integration of Scientific Research, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant 2022XD001.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 364–376, 2022.
https://doi.org/10.1007/978-3-031-19208-1_30

## 1 Introduction

With the development of cloud services, a great number of organizations and companies apply cloud computing technologies to their businesses, which leads the limited resources to be utilized efficiently and the hardware overhead to be reduced [1]. Meanwhile, various industries generate a large amounts of data, which contains an enormous value and causes complicated data management problems at the same time [2]. In order to relieve the burden on local data management and system maintenance, data owners naturally store the massive data in the cloud server and process them through cloud computing [3]. Cloud storage and sharing techniques have become a significant way of data interaction, but then it can cause security problems such as the privacy leakage of users [4] and the difficulty in managing shared data [5]. So data has to be encrypted before uploading to the public cloud. Symmetric encryption can be applied to large amounts of data encryption to ensure efficiency and data confidentiality. After the shared data is encrypted, how to share the encrypted data with the authenticated data users is also a critical issue for cloud storage.

To meet the secure requirements of outsourced data, attribute-based encryption (ABE) is envisioned to be a promising cryptographic primitive for protecting the data security and realizing the fine-grained access control in data sharing. Attribute-based encryption (ABE) is one of the most promising methods to ensure the confidentiality and fine-grained access control simultaneously, which was first proposed by Sahai et al. in 2005 [6]. Ciphertext-policy ABE (CP-ABE) is a derivative encryption algorithm of ABE [7]. In CP-ABE, a data owner defines an access structure to encrypt its data which decides attributes that a user needs for decryption. Furthermore, in order to realize the efficient using of data ciphertext, searchable encryption techniques enable data users to retrieve ciphertext data by using keyword index search without revealing the keywords [8].

Nevertheless, most of the data sharing scheme in the cloud storage rely on third parties, which may cause data stealing, leaking, tampering, or misused in case of attack or lack of monitoring [9]. In [10], Zhang et al. proposed a scheme called secure door on cloud to protect the data from being leaked, which can also effectively prevent third-parties from stealing plaintext data. Blockchain is a specific data structure that combines data blocks into chains in chronological order, which ensures its immutability and unforgeability [11]. The distributed verification feature of blockchain ensures complete data variability and tamper-proof. However, blockchain-based transaction models often lack the protection of users' privacy. While blockchain provides public audit and traceability of transaction data, it also exposes the private information of both parties of the transaction, and some private data may be used illegally [12], which obviously does not meet the practical needs of privacy protection.

To solve this problem, some studies combine cryptography to protect privacy indirectly by hiding the connection between data and users. For example, Guan et al. build a privacy-preserving blockchain energy trading scheme, which can achieve fine-grained access control through transaction arbitration in the ciphertext form [13]. Shen et al. proposed a smart contract-enabled three-party collaboration model for data sharing in multiple clouds [14]. In [15], Wang et al.

constructed a data-sharing system over lattice by designing an efficient identity-based broadcast encryption scheme, which achieves the data confidentiality and the identity privacy, simultaneously. However, these existing work cannot guarantee the hierarchical management of data and meet the needs of data sharing at different levels. In addition, cloud storage combined with the blockchain not only solves the problem of the limited storage capacity of the blockchain, but also improves throughput and enhances scalability.

So our goal is to design a hierarchical management scheme for shared data and privacy data based on blockchain and cloud storage according to existing work. In our scheme, privacy data and shared data are encrypted in a hierarchical manner. And then different levels of encrypted data can be decrypted according to the hierarchical authority of the data user. Therefore, the data sharing of different needs is achieved while ensuring privacy. The access policy used in this paper is Linear Secret Sharing Scheme (LSSS) [16], which not only enables fine-grained access control, but also has high computational efficiency.

The main contributions of our scheme can be summarized as follows:

- We propose a new hierarchical ciphertext policy attribute-based encryption scheme for cloud data sharing based on blockchain, which realizes the separation of the privacy part and the shared part of the data. The attribute policy is embedded in the hierarchical ciphertext, which can achieve hierarchical controllable data sharing.
- We store the data ciphertext in the cloud and key ciphertext in the blockchain, which guarantees the data integrity in cloud and the keywords tamper-proof on blockchain. Therefore, it reduces the risk of data leakage.
- We combine the searchable encryption technology in our scheme, which can realizes secure search on blockchain. The symmetric encryption combined with the searchable CP-ABE not only ensures the storage security of data in the cloud but also achieves the efficiency and convenience of data searching.

## 2 Preliminaries

### 2.1 Bilinear Maps

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cylic groups with prime order $q$. Let $g$ be a generator of $\mathbb{G}$ and $e$ be a bilinear map, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ with the following three properties:

1) Bilinearity: $\forall u, v \in \mathbb{G}$, and $a, b \in \mathbb{Z}_q^*$, we have $e\left(u^a, v^b\right) = e\left(u, v\right)^{ab}$, where $\mathbb{Z}_q^*$ is in the integers modulo $q$.
2) Nondegenerate: $e(g, g) \neq 1$.
3) Computable: $\forall u, v \in \mathbb{G}$, there is an efficient algorithm to compute $e(u, v)$.

### 2.2 Access Structure

Let $P = \{P_1, P_2, ..., P_n\}$ be a set of parties, a collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, ..., P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure is a collection $\mathbb{A}$ of non-empty subsets of $P = \{P_1, P_2, ..., P_n\}$, such as $\mathbb{A} \subseteq 2^{\{P_1, P_2, ..., P_n\}} \backslash \emptyset$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets.

### 2.3 Linear Secret Sharing Schemes (LSSS)

A secret sharing scheme $\Pi$ over a set of parties is called linear if

1) The shares of all parties form a vector over $\mathbb{Z}_q^*$.
2) There is a matrix $M$ with $l$ rows and $n$ columns, which is called the share generating matrix for $\Pi$. For all $i = 1, 2, ..., l$, in the $i - th$ row of $M$, we will use the function $P(i)$ as the row label. A column vector $\overrightarrow{v} = (s, r_2, ..., r_n)$ is generated, where $s \in \mathbb{Z}_q^*$ is the secret to be shared and $r_2, ..., r_n \in \mathbb{Z}_q^*$ are randomly chosen, then $M \cdot \overrightarrow{v}$ is the vector of $l$ shares of the secret $s$ according to $\Pi$. The share $(M \cdot \overrightarrow{v})_i$ belongs to party $P(i)$.

## 3 System Model

The system model of BC-HSABE is depicted in Fig. 1, which mainly includes five entities, namely attribute authority (**AA**), public cloud server (**CS**), blockchain (**BC**), data owner (**DO**) and data user (**DU**). In the following, we will introduce the features of these five entities in detail.

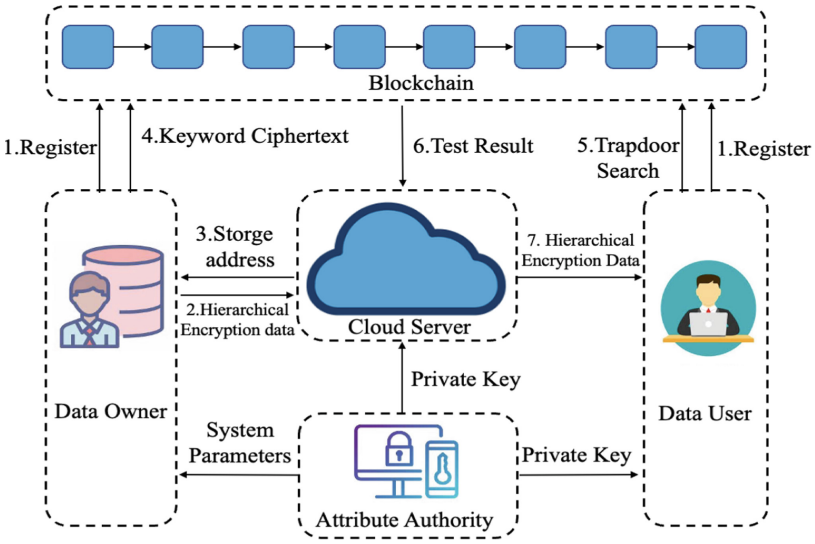



**Fig. 1.** System model of BC-HSABE

1) Attribute Authority (**AA**): AA is a trusted organization that mainly responsible for generating system parameters, system master keys, and private keys for DU and CS. When DU joins the system, AA assigns it a unique identifier $UID$ and an attribute set $S_{uid}$.

2) Cloud Server (**CS**): CS is responsible for storing the encrypted data file provided by DO and returning the file storage address $F_{id}$ to DO's account on the blockchain. If a data requester DU meets the access policy and the searching keyword test results are verified successfully, CS sends the corresponding encrypted data file to DU's account on the blockchain.
3) Blockchain (**BC**): BC is responsible for verifying the transaction blocks provided by DO. In this paper, BC adopts PBFT consensus algorithm, which ensures the consistency of the distributed node network under the condition that no more than 1/3 of the malicious nodes in the whole network. In addition, BC is responsible for searching the ciphertext keyword trapdoor provided by DU. When the search is successful, BC sends the verification result to the CS, and CS returns the encrypted data files to DU.
4) Data Owner (**DO**): DO extracts keywords from data files. Then DO encrypts data files $F_1$, $F_2$ hierarchically using symmetric keys $k_1$ and $k_2$ to get corresponding encrypted data files $C_{F_1}$, $C_{F_2}$, and then establishes keyword indexes. DO defines the access policy and encrypts the symmetric keys $k_1$ and $k_2$ under the access policy to obtain the corresponding ciphertext $C_1$ and $C_2$. Finally, DO uploads the encrypted data files $C_{F_1}, C_{F_2}$ and data ciphertext $C_1$, $C_2$ to CS, and uploads the transaction consisting of keyword ciphertext and file storage address $F_{id}$ to BC to form a new block.
5) Date User (**DU**): After the searched trapdoor generated by the DU is verified, DU obtains the encrypted data ciphertext $C_1$, $C_2$. DU uses his own private key to decrypt the ciphertext $C_2$ to get the symmetric key $k_2$, which enables it to decrypt the encrypted second-level data file $C_{F_2}$. If DU is interested in the detailed first-level data, he will directly interacts with DO and gets permission to obtain the parameters and then decrypt the first-level file $C_{F_1}$.

## 4 Construction

In this section, we present the detailed construction of BC-HSABE. The overview procedure of our BC-HSABE can be divided into six phases as shown in Fig. 2, named setup phase, key generation phase, encryption phase, trapdoor phase, test phase and decryption phase. The specific implementation process of these six stages are described below.

(1) **System Setup**
The setup algorithm is performed by AA. It takes a security parameter $\lambda$ as input and outputs the public parameters $PP$ and master key $MSK$.
1) Select two cyclic groups of prime order q: $\mathbb{G}$ with generator $g$ and $\mathbb{G}_T$.
2) Select a map function $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.
3) Choose two hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}$, $H_2 : \{0,1\}^* \rightarrow \mathbb{G}$.
4) AA defines a set of attributes $U$, each attribute $x \in U$.
5) Randomly choose $\alpha, \beta, a, b, c \in Z_q^*, h_1, ..., h_U \in \mathbb{G}$.
Finally, the generated master key $MSK = \{\alpha, \beta, a, b, c\}$, and AA published the public parameters $PP = \{g, e(g,g)^{\alpha}, e(g,g)^{b}, g^{a}, g^{\beta}, g^{c}, g^{(1/c)}, h_1, ..., h_U\}$.

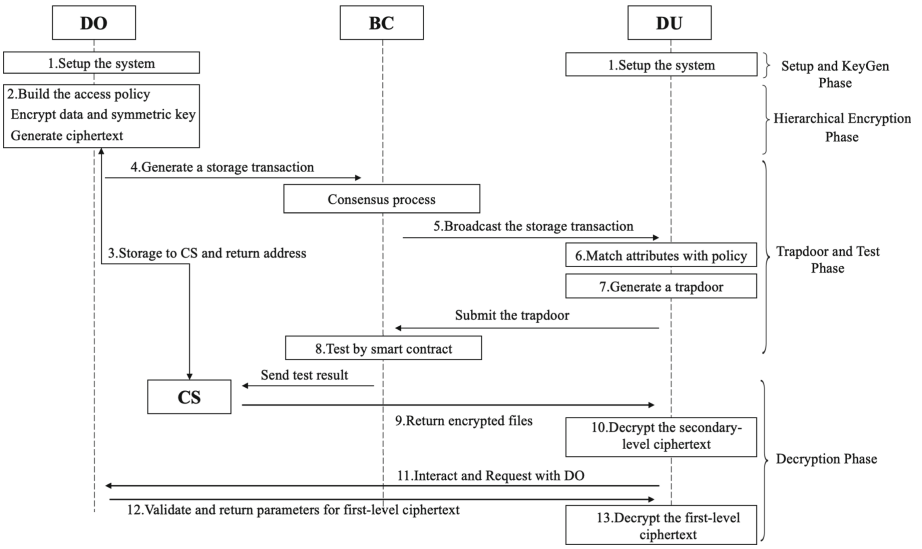



**Fig. 2.** The overview procedure of BA-HSABE

(2) **Key Generation**
When DU joins the system, AA assigns a unique identifier $UID$ and an attribute set $S_{uid}$ to DU. AA randomly selects $t \in Z_q^*$ and performs the following calculations to generate a private key for DU. Subsequently, AA sents $\{K = g^{\alpha} g^{at+b}, L = g^t, x \in S, K_x = h_x^t\}$ to the corresponding user node.

(3) **Hierarchical Encryption**
This phase is divided into two parts: data encryption and keyword encryption. As a transaction initiator, DO formulates access policy and divides the data into two levels: private data and shared data according to the degree of importance. The first-level data file $F_1$ contains the detailed and confidential data. The secondary-level data file $F_2$ includes the basic shared data.

1) Part 1: Data Encryption
For two levels of data files $F_1, F_2$, DO randomly selects the corresponding symmetric keys $k_1$ and $k_2$, and uses symmetric encryption algorithm to encrypt data files $F_1, F_2$ to obtain $C_{F_1} = E_{k_1}(F_1), C_{F_2} = E_{k_2}(F_2)$.

2) Part 2: Keyword Encryption
DO extracts the keyword $\omega$ from the data files $F_1, F_2$. Subsequently, DO formulates an access strategy and generates an LSSS access structure such as $(M, \rho)$, where function $\rho$ represents the position of a given attribute in the matrix $M$, and $\rho(i) \in \{Att_1, Att_2, ..., Att_U\}$. $M$ is a $l \times n$ matrix, where $l$ represents the number of attributes involved in the access policy and $n$ represents the variable defined by the LSSS conversion method. The rows of $M$ represent different attribute categories, and the columns are variables defined by the attribute values. In the initial stage of encryption,

we randomly select the vector $\overrightarrow{v} = (s, y_2, y_3, ..., y_n) \in Z_q^*$. For each row in the matrix, a new parameter $\lambda_i = M_i \cdot \overrightarrow{v}$ is calculated. After DO randomly selects $s, d_0, r_1, r_2, ..., r_l \in Z_q^*$, the two ciphertexts of the symmetric keys are calculated as follows:

$$C_1 = k_1 \cdot e(g,g)^{\alpha s}, C_2 = k_2 \cdot e(g,g)^{\alpha s} / e(g,g)^{d_0 s} = k_2 \cdot e(g,g)^{(\alpha - d_0)s},$$
$$C' = g^s, B = e\left(g^{as}, g^{\beta}\right), \forall 1 \le i \le l, C_i = g^{a\lambda_i} h_{\rho_i}^{-r_i}, C_i' = g^{a\lambda_i},$$
$$B_i = \left[(H_1(\omega) \cdot H_2(\rho(i))\right]^{\lambda_i}, D_i = g^{r_i}.$$

DO uploads $(C_1, C_2, C', C_{F_1}, C_{F_2})$ to CS, and gets the corresponding storage address $F_{id}$ returned by CS. Then DO forms $TX = (B, B_i, C_i, C_i', D_i)$ as a transaction order on the block and signs $TX$ to get the corresponding signature $\delta$. Then DO submits an authentication request to the master nodes of blockchain, and the master nodes execute the $PBFT$ consensus algorithm to verify the transaction. If the number of correct verification results is greater than $f+1$, a new block has been added to the blockchain, where $f$ represents the number of malicious nodes and the total number of nodes is $N > 3f+1$.

(4) **Trapdoor Generation**
DU generates a trapdoor of the interested keyword $\omega'$. For each attribute $a_i$ in $S_{uid}$, DU randomly selects $r_i \in Z_q^*$ and then calculates: $T_i = [H_1(\omega') \cdot H_2(a_i)]r_i, T_i' = g^{ar_i}$. Subsequently, DU uploads the triplet $(UID, T_i, T_i')$ to BC.

(5) **Test**
When the keyword $\omega'$ searched by DU and the attribute set $S_{uid}$ possessed by DU satisfy the access structure $(M, \rho)$, the blockchain node performs the following operations: select $I \subset \{1, 2, ..., l\}$, and define $I = \{i : \rho(i) \in S\}$. Then, according to the LSSS protocol, a set of constants $\{\omega_i \in Z_q^*\}_{i \in I}$ can be found in polynomial time, such that $\sum_{i \in I} \omega_i \cdot \lambda_i = s$. Subsequently, the consensus nodes perform the following test procedure:

$$\frac{\prod_{i \in I} e\left(g^{\beta} \cdot T_i, C_i'\right)^{\omega_i}}{\prod_{i \in I} e\left(T_i', B_i\right)^{\omega_i}} = B \tag{1}$$

If the Eq. (1) is not true, the error symbol "$\perp$" is printed. If the equation holds, the blockchain node sends verification result and DU's account on the blockchain to the CS according to the data storage address $F_{id}$.

(6) **Decryption**
CS sends the two levels encrypted data ciphertext to the account of DU which has verified during the **Test** phase. However, DU can only decrypt secondary-level ciphertext using his private key. If interested, the DU that meet the conditions can apply for the corresponding detailed data first-level ciphertext. After the identity of DU have been verified by the DO, the DU will receive the relevant parameters for decrypting the first-level ciphertext.

1) Decrypt the secondary-level ciphertext: After DU receives the data ciphertext returned from CS, DU performs the following decryption calculation:

$$S_1 = \frac{e\left(C', K\right)}{\prod_{i \in S}\left(e\left(L, C_i\right) e\left(D_i, K_x\right) e(g,g)^{(d_0+b)\lambda_i}\right)^{\omega_i}} = e(g,g)^{(\alpha - d_0)s} \tag{2}$$

Then the symmetric encryption key $k_2$ and the secondary-level file $F_2$ can be obtained by the calculation: $k_2 = \frac{C_2}{S_1} = \frac{k_2 \cdot e(g,g)^{(\alpha-d_0)s}}{e(g,g)^{(\alpha-d_0)s}}, F_2 = Dec_{k_2}(F_2)$.

2) Decrypt the first-level ciphertext: When DU wants to learn the content of the first-level file, he will directly interact with the transaction initiator DO. After the identity of DU is confirmed and verified by DO, DU will receive $g^{d_0}$ provided by DO and performs the calculations to obtain the first-level data file $F_1$: $k_1 = \frac{C_1}{S_1 \cdot e(g^s, g^{d_0})} = \frac{k_1 \cdot e(g,g)^{\alpha s}}{e(g,g)^{(\alpha-d_0)s} \cdot e(g,g)^{d_0 s}}, F_1 = Dec_{k_1}(F_1)$

## 5 Security Analysis

In this section, we conduct the security analysis of BC-HSABE from three aspects: data security, tamper-proofing and privacy-preservation.

### 5.1 Data Security

BC-HSABE can guarantee the confidentiality of the shared data. Before uploading the shared data to the public cloud, we use the symmetric encryption algorithm to encrypt the data, and then we encrypt the symmetric key by ABE algorithm. So the malicious attackers cannot get any information without the authority attributes, which guarantees the confidentiality of the shared data.

### 5.2 Tamper-Proofing

Our BC-HSABE uses the distributed blockchain in the data sharing framework, which can resist single-point attacks. We encrypt the hierarchical data file and store them in the cloud, keyword index is stored in the blockchain. In addition, the blockchain uses cryptographic primitives such as public key systems, hash calculations, and digital signatures to conduct transactions in an anonymous manner, which can ensure the integrity and usability of keyword trapdoor.

### 5.3 Privacy-Preservation

BC-HSABE realizes the protection of users' privacy during the whole process of data sharing. We designed and implemented a fine-grained access control scheme with hierarchical encryption to effectively protect the privacy information of the data owner. The design of the two-level ciphertext ensures that only the data requester who satisfy the access policy can decrypt the basic shared data information and interact directly with data owner to apply for the relevant parameters for decrypting first-level ciphertext.

**Table 1.** The comparison of computational performance

| Schemes | Setup | KeyGen | Encryption | Trapdoor | Test | Decryption |
|---|---|---|---|---|---|---|
| Scheme [17] | $P + (2N_x + 3)E + E_T$ | $(2N_x + 4)E$ | $E_T + (2l + 4)E$ | – | – | $(2N_x + 1)P + N_x E_T$ |
| Scheme [18] | $P + (N_x + 2)E + E_T$ | $(N_x + 4)E$ | $E_T + (3l + 1)E$ | $E$ | $lP + E$ | $(2N_x + 2)P + 2N_x E_T$ |
| Scheme [19] | $P + 4E$ | $(N_x + 2)E$ | $P+(4l+6)E$ | $(N_x + 5)E$ | $(2N_x + 3)P + N_x E$ | $(2N_x + 1)P + N_x E$ |
| BC-HSABE | $4E+2E_T+P$ | $(3 + N_x)E$ | $2Sym + 2E_T + (4l + 2)E + P$ | $N_x E$ | $2lP + 2lE_T$ | $(2N_x + 1)P + 2N_x E_T + Sym$ |

## 6 Performance Evaluation

### 6.1 Theoretical Analysis and Comparison

In this section, we give the theoretical performance analysis from the perspective of computation cost, and make some comparison with schemes in [17–19].

The theoretical evaluation of the computation amount of the key operations in our BC-HSABE is shown in Table 1, where $P$ represents the pairing operation, $E$ represents the group exponentiation in $\mathbb{G}$, $E_T$ represents the group exponentiation in $\mathbb{G}_T$, $N_x$ represents the number of attributes a user possesses, $l$ represents the number of attributes embedded in a ciphertext and $Sym$ represents the operations required for symmetric encryption/decryption.

In the Setup phase and Key Generation phase, the computation amount in BC-HSABE is smaller than the scheme in [17,18]. In the Test phase the computation cost of BC-HSABE is greater than scheme in [18], but it is lower than scheme in [19]. This is because the trapdoor generation of scheme [18] does not involve user's attributes, which reduces the computational overhead but increases the security risk. In the Encryption phase, the computation cost of BC-HSABE is greater than the scheme in [17,18], while it is similar to the scheme in [19]. In the Decryption phase, the computation cost of BC-HSABE is greater than that of the scheme in [17,19], while it is similar to the scheme in [18]. Since BC-HSABE needs to implement two-levels of ciphertexts hierarchical encryption and decryption, so a slightly higher computational cost is required.

### 6.2 Numerical Experimental Analysis

In this section, we present the experimental evaluation results of our BC-HSABE performance. This section carries out numerical simulation experiment on the scheme algorithm. The numerical simulation experiment was carried out in Mac operating system using PBC (Pairing-Based Cryptography) library of C language. We run simulation experiments in Clion of MacBook Air (M1, 2020, 8 GB RAM) and analyze the computational efficiency of the BC-HSABE and Wang's scheme in [18] by changing the number of attributes. The number $n$ of attributes is 4, 6, 8, 10, 12, 14, 16, 18, 20, respectively. The experimental result is the average of the algorithm running 50 times, as shown in Fig. 3.

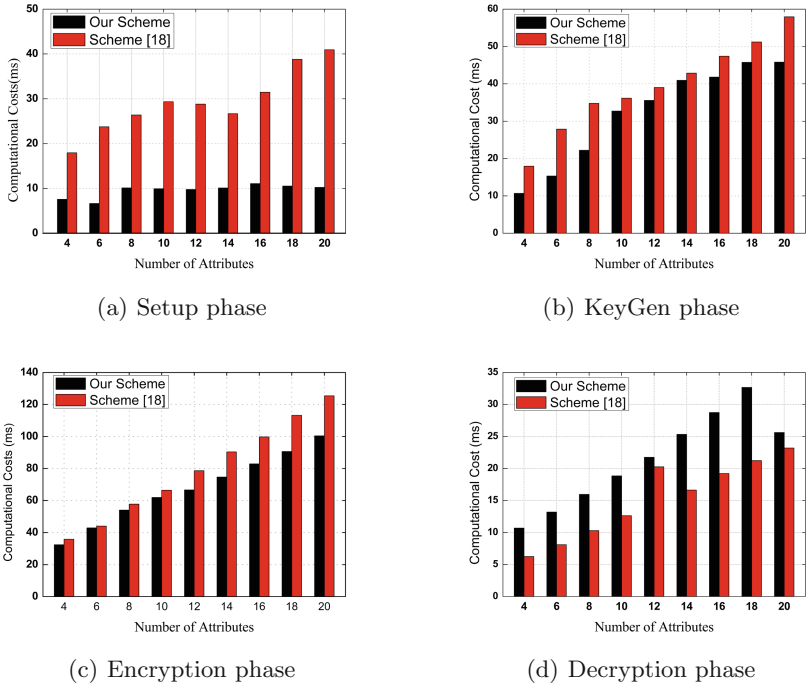



**Fig. 3.** Relationship between the number of attributes and the computational costs in different phase

Figure 3(a) shows the time costs of Setup phase in two attribute-based encryption schemes, where we can see that time cost of scheme in [18] is linear to the number of attributes. However, it is almost no change as the number of attributes increases in BC-HSABE. As we can see from Fig. 3(b) and (c), the time costs of KeyGen phase and Encryption phase in scheme [18] and BC-HSABE are linear to the number of attributes while the efficiency of BC-HSABE is higher than scheme in [18] overall these two phase. The time costs of decryption are presented in Fig. 3(d), it is clear that both results of these two schemes are linear to the number of attributes while our BC-HSABE needs more time cost for decryption. Since BC-HSABE implements two-levels of ciphertexts hierarchical decryption, we calculate the total time which includes decryption of two hierarchical level ciphertexts. It is worthwhile to spend extra time implementing ciphertext hierarchical management under privacy protection.

## 7 Conclusion

In this paper, we introduce a blockchain-aied data sharing scheme with privacy protection and hierarchical ciphertext access control, named BC-HSABE. By using attribute-based encryption technology, we encrypt shared data and private data hierarchically. Searchable encryption technology is adopted to achieve fine-grained access control and keyword ciphertext search. Our BC-HSABE realizes

the dynamic rights management based on attributes. The data owner packages the keyword index into transactions, which are stored on the blockchain after distributed verification by blockchain nodes. The searching process is carried out on the blockchain, which ensures the security of the keywords. The security analysis and experimental analysis show that our scheme is safe and effective.

## A Appendix

1. **Correctness of Eq. (1)**

$$\begin{aligned}
&\frac{\prod_{i\in I} e\left(g^{\beta}\cdot T_i, C_i'\right)^{\omega_i}}{\prod_{i\in I} e\left(T_i', C_i\right)^{\omega_i}} = \frac{\prod_{i\in I} e\left(g^{\beta}\cdot\left[H_1\left(w'\right)\cdot H_2\left(a_i\right)\right]^{r_i}, \left(g^{a}\right)^{\lambda_i}\right)^{\omega_i}}{\prod_{i\in I} e\left(\left(g^{a}\right)^{r_i}, \left[H_1(w)\cdot H_2(\rho(i))\right]^{\lambda_i}\right)^{\omega_i}} \\
&= \frac{\prod_{i\in I} e\left(\left[H_1\left(w'\right)\cdot H_2\left(a_i\right)\right]^{r_i}, g^{a\lambda_i\omega_i}\right)\prod_{i\in I} e\left(g^{\beta}, g^{a\lambda_i\omega_i}\right)}{\prod_{i\in I} e\left(g^{a\omega_i\lambda_i}, \left[H_1(w)\cdot H_2(\rho(i))\right]^{r_i}\right)} \\
&= \prod_{i\in I} e\left(g^{\beta}, g^{a\lambda_i\omega_i}\right) \\
&= e\left(g^{\beta}, g^{as}\right) = B
\end{aligned}$$

2. **Correctness of Eq. (2)**

$$\begin{aligned}
S_1 &= \frac{e\left(C', K\right)}{\prod_{i\in S}\left(e\left(L, C_i\right) e\left(D_i, K_x\right) e(g,g)^{(d_0+b)\lambda_i}\right)^{\omega_i}} \\
&= \frac{e\left(g^{s}, g^{\alpha} g^{at+b}\right)}{\prod_{i\in S}\left(e\left(g^{t}, g^{a\lambda_i} h_{\rho(i)}^{-r(i)}\right) e\left(g^{r(i)}, h_x^{t}\right) e(g,g)^{(d_0+b)\lambda_i}\right)^{\omega_i}} \\
&= \frac{e(g,g)^{\alpha s} e(g,g)^{ats} e(g,g)^{bs}}{\prod_{i\in S}\left(e(g,g)^{at\lambda_i} e(g,g)^{(d_0+b)\lambda_i}\right)^{\omega_i}} \\
&= \frac{e(g,g)^{\alpha s} e(g,g)^{ats} e(g,g)^{bs}}{e(g,g)^{ats} e(g,g)^{(d_0+b)s}} \\
&= e(g,g)^{(\alpha-d_0)s}
\end{aligned}$$

## References

1. Cai, Z., Zheng, X.: A private and efficient mechanism for data uploading in smart cyber-physical systems. IEEE Trans. Netw. Sci. Eng. **7**(2), 766–775 (2020). https://doi.org/10.1109/TNSE.2018.2830307
2. Zheng, X., Cai, Z.: Privacy-preserved data sharing towards multiple parties in Industrial IoTs. IEEE J. Sel. Areas Commun. **38**(5), 968–979 (2020). https://doi.org/10.1109/JSAC.2020.2980802
3. Zhu, L., Dong, H., Shen, M., Gai, K.: An incentive mechanism using shapley value for blockchain-based medical data sharing. In: 2019 IEEE 5th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International

Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), pp. 113–118 (2019). https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2019.00030
4. Cao, S., Wang, J., Du, X., Zhang, X., Qin, X.: CEPS: a cross-blockchain based electronic health records privacy-preserving scheme. In: ICC 2020–2020 IEEE International Conference on Communications (ICC), pp. 1–6 (2020). https://doi.org/10.1109/ICC40277.2020.9149326
5. Zhang, J., Yang, Y., Liu, X., Ma, J.: An efficient blockchain-based hierarchical data sharing for healthcare Internet of Things. IEEE Trans. Ind. Inform. **18**(10), 7139–7150 (2022). https://doi.org/10.1109/TII.2022.3145851
6. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
7. Roy, S., Chuah, M.: Secure data retrieval based on ciphertext policy attribute-based encryption (CP-ABE) system for the DTNs. Technical report, Citeseer (2009)
8. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_30
9. Alatawi, S., Alhasani, A., Alfaidi, S., Albalawi, M., Almutairi, S.M.: A survey on cloud security issues and solution. In: 2020 International Conference on Computing and Information Technology (ICCIT-1441), pp. 1–5 (2020). https://doi.org/10.1109/ICCIT-144147971.2020.9214397
10. Zhang, H., Fang, L., Jiang, K., Zhang, W., Li, M., Zhou, L.: Secure door on cloud: a secure data transmission scheme to protect Kafka's data. In: 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), pp. 406–413 (2020). https://doi.org/10.1109/ICPADS51040.2020.00061
11. Yan, B., Yang, Z., Ren, Y., Tan, X., Liu, E.: Microblog sentiment classification using parallel SVM in apache spark. In: 2017 IEEE International Congress on Big Data (BigData Congress), pp. 282–288 (2017). https://doi.org/10.1109/BigDataCongress.2017.43
12. Zhaofeng, M., Lingyun, W., Xiaochang, W., Zhen, W., Weizhe, Z.: Blockchain-enabled decentralized trust management and secure usage control of IoT big data. IEEE Internet Things J. **7**(5), 4000–4015 (2019)
13. Guan, Z., Lu, X., Yang, W., Wu, L., Wang, N., Zhang, Z.: Achieving efficient and privacy-preserving energy trading based on blockchain and ABE in smart grid. J. Parallel Distrib. Comput. **147**, 34–45 (2021)
14. Shen, M., Duan, J., Zhu, L., Zhang, J., Du, X., Guizani, M.: Blockchain-based incentives for secure and collaborative data sharing in multiple clouds. IEEE J. Sel. Areas Commun. **38**(6), 1229–1241 (2020)
15. Wang, F., Wang, J., Shi, S.: Efficient data sharing with privacy preservation over lattices for secure cloud storage. IEEE Syst. J. **16**(2), 2507–2517 (2021). https://doi.org/10.1109/JSYST.2021.3077236
16. Cramer, R., Damgård, I., Maurer, U.: General secure multi-party computation from any linear secret-sharing scheme. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 316–334. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_22
17. Yang, K., Jia, X.: ABAC: attribute-based access control. In: Yang, K., Jia, X. (eds.) Security for cloud storage systems, pp. 39–58. Springer, New York (2014). https://doi.org/10.1007/978-1-4614-7873-7_3

18. Wang, S., Zhang, D., Zhang, Y., Liu, L.: Efficiently revocable and searchable attribute-based encryption scheme for mobile cloud storage. IEEE Access **6**, 30444–30457 (2018)
19. Yanli, C., Minhui, Z.: Privacy protection attribute-based ciphertext search scheme. Appl. Res. Comput. **36**(4), 1158–1164 (2019)

# Efficient Certificateless Ring Signcryption Scheme with Constant Ciphertext Length on Blockchain

Yan Jin[1,2], Chunxiao Ye[1,2(✉)], Mengqing Yang[1,2], and Chunming Ye[1,2]

[1] College of Computer Science, Chongqing University, Chongqing, China
jiny@cqu.edu.cn, yangmengqing@pku.edu.cn, ccxmye@163.com
[2] Key Laboratory of CPS-DSC, MoE, Chongqing University, Chongqing, China
yecx@cqu.edu.cn

**Abstract.** Ring signcryption schemes have been proposed on blockchain, but compared with ring signcryption using public key infrastructure (PKI), certificateless can simplify the certificate management process. At the same time, the length of the ciphertext increases with the size of the ring is also a pressing challenge to be solved. In this paper, we propose a new certificateless ring signcryption scheme with constant ciphertext length and prove the security under the random oracle model. Compared with other schemes, the computational overhead of this scheme is smaller and more efficient.

**Keywords:** Blockchain · Ring signcryption · Certificateless · Constant ciphertext length

## 1 Introduction

Blockchain as an emerging technology has proven its feasibility in a wide range of applications due to its decentralized, tamper-proof, and anonymity properties [20]. Also ring signature as a powerful cryptographic protocol can be used to achieve anonymity, the concept of which was first proposed by Rivest et al. [13]. In Monero, a modified version of the ring signature proposed by Fujisaki et al. [6] would be used to ensure untraceability.

Ring signature allow a signer can arbitrarily choose some members to form a ring and generate a ring signature without the assistance of other ring members. Any verifier can know that the message is from a member of the ring, but does not know exactly who the signer is. However, the signature size of the ring signature increases with the ring size. To solve this problem, Dodis et al. [5] proposed the first constant size ring signature scheme. Chandran et al. [3] gave a sublinear size ring signature without random oracles and showed some drawbacks of the signature size. Khuc et al. [10] propose a more efficient unique ring signature of logarithmic size on blockchain and give proof of security. RingCT [12] was adopted late in Monero, which increased the number of hash but made the signature length reduced by half. Because the shortened signature not only reduces


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 377–388, 2022.
https://doi.org/10.1007/978-3-031-19208-1_31

the network load, it also reduces the size of the transaction, which reduces the transaction fee for the mechanism that calculates the transaction fee by the number of bytes [17].

To improve efficiency, ring signature were extended to ring signcryption [9]. For the ring signcryption scheme in traditional public key infrastructure (PKI) [1], the signcrypter must first check the certificates of all ring members before it can generate a ring ciphertext on behalf of the ring. If there is an extreme case where the certificates of ring members are invalid, the anonymity of the signcrypter may be at risk. Similarly, for the verifier, the same check must be performed before unsigncryption. This would lead to inefficiencies in the overall scheme, as the computational cost would increase linearly with the ring size. Removing the public key certificate simplifies the process of joining and revoking ring members. While the use of identity-based encryption (IBE) [14] can eliminate this costly verification, there is a key escrow problem. Therefore, using certificateless [2] reduces the risk of private key management and the pain of joining and revoking ring members.

Wang et al. [18] constructed a certificateless ring signcryption scheme, which proved to be secure. However, their scheme requires $3n + 5$ pairing operations, and the overhead of bilinear pairing operations is too large. Sharma et al. [15] proposed a pairing-free certificateless ring signcryption scheme for wireless sensor networks, but Shen et al. [16] presented two specific attacks to demonstrate that their scheme provides neither confidentiality nor unforgeability against type $I$ adversary. Zhang et al. [21] proposed a certificateless ring signcryption scheme for protecting user privacy in the smart grid, which eliminates bilinear pairing and exponentiation computation costs by using modular multiplication on elliptic curves. Guo et al. [7] proposed a certificateless ring signcryption scheme from pairings and optimizes the efficiency in this scheme so that only one bilinear pairing operation is required for signcryption and three bilinear pairing operations are required for unsigncryption. Zhao et al. [22] proposed an authenticated certificateless ring signcryption scheme to solve the problems of vehicle user privacy information protection and communication message transmission security. In order to resist quantum computing attacks, Yu et al. [19] proposed a certificateless multivariable ring signcryption scheme with anti-quantum. Guo et al. [8] in consideration of the security on the vehicular ad hoc networks, chose to use the certificateless ring signcryption scheme to realize conditional privacy.

With the increasing prosperity of blockchain, there is a desire to improve the performance of ring signcryption schemes. Improving the performance of ring signcryption by reducing the size of the ring signcryption and the signcryption scheme overhead has been an important research direction.

In this paper, we propose a new certificateless ring signcryption scheme that possesses a constant ciphertext length and does not grow longer as the ring size increases. This scheme has confidentiality, unforgeability, and anonymity, and is proven to be secure under the random oracle model. Compared with other schemes, this scheme has a shorter ciphertext length and smaller computational overhead.

The paper is organized as follows. Section 2 introduces the hard problem, the certificateless ring signcryption model, and the security model. Section 3 specifies the implementation of the certificateless ring signcryption scheme. Section 4 analyzes the security of the scheme. Section 5 simulates the scheme and performs performance analysis. Finally, the paper is summarized in Sect. 6.

## 2 Preliminaries

### 2.1 Hard Problems

**Definition 1.** Given a group $G_p$ of prime order $p$, $P$ is a generator of $G_p$. The computational Diffie-Hellman problem (CDHP) is given $(P, aP, bP)$ to compute $abP$, where $a, b \in Z_p^*$.

**Definition 2.** Given a group $G_p$ of prime order $p$, $P$ is a generator of $G_p$. The discrete logarithm problem (DLP) is given $(P, aP)$ to compute $a$, where $a \in Z_p^*$.

### 2.2 Certificateless Ring Signcryption Model

The certificateless ring signcryption scheme consists of six algorithms, which are described in detail as follows:

**Setup.** Input a security parameter $k$, KGC publishes the generated public parameters *params* and secretly saves the generated system master key $s$.

**Set Partial Private Key.** Input the user's identity $ID_A$ and $U_A$, the system master key $s$ and the public parameters *params*, KGC generates the user's partial private key $d_A$ and partial public key $Y_A$.

**Set Public Key.** Input the user's identity $ID_A$, $U_A$, partial public key $Y_A$ and the public parameters *params*, the user generates its own public key $PK_A$.

**Set Private Key.** Input the user's identity $ID_A$, $U_A$, partial private key $d_A$ and the public parameters *params*, the user generates its own private key $SK_A$.

**Signcryption.** Input the public key $PK_B$ of the receiver, a public key ring $RL$ including the public keys of $n-1$ other users selected by the sender and its public key, the private key $SK_A$ of the sender, the public parameters *params* and the message $m$, the sender generates the ciphertext $\sigma$.

**Unsigncryption.** Input ciphertext $\sigma$, the receiver's private key $SK_B$, a public key ring $RL$, and the public parameters *params*, the receiver generates the message $m$.

### 2.3 Security Model

Type $I$ adversary $\Im_I$ cannot obtain the system master key but can replace any user's public key with a value of its choosing. Type $II$ adversary $\Im_{II}$ can obtain the master key, but cannot replace the user's public key. The confidentiality and unforgeability of the scheme in this paper are defined by games in which the adversary $\Im$ can access the following oracle through the challenger $C$:

- Partial private key queries: $\Im$ sends the $ID_i$ to $C$, $C$ returns the partial private key $d_i$.
- Public key queries: $\Im$ sends the $ID_i$ to $C$, $C$ returns the corresponding public key $PK_i$.
- Private key queries: $\Im$ sends the $ID_i$ to $C$, $C$ returns the corresponding private key $SK_i$.
- Public key replacement queries: $\Im$ sends the replaced public key $PK_i'$ and $ID_i$ to $C$, $C$ replaces the public key $PK_i'$ corresponding to $ID_i$ with $PK_i'$. $\Im_{II}$ cannot perform the query of replacing the public key.
- Signcryption queries: $\Im$ sends the message $m$, $n$ potential senders to form a public key ring $RL$, and the receiver's $ID_j$ to $C$. $C$ returns the ciphertext $\sigma$.
- Unsigncryption queries: $\Im$ sends the ciphertext $\sigma$, the public key ring $RL$, and the receiver's $ID_j$ to $C$. $C$ returns the message $m$.

**Definition 3.** The scheme is said to have indistinguishability against adaptive chosen ciphertext attack (IND-CCA2) if adversary $\Im$ does not have a non-negligible probability advantage in the confidentiality game.

In the game, adversary $\Im_I$ cannot query the private key and cannot request the private key if the corresponding public key has been replaced unless $\Im_I$ submits the corresponding secret value to $C$. $\Im_I$ cannot query both the public key, which replaced the target identity before the challenge phase and the signcryption or partial private key. $\Im_I$ cannot perform an unsigncryption query on the target ciphertext.

Then, adversary $\Im_{II}$ cannot query the private key and replace the public key. $\Im_{II}$ cannot perform an unsigncryption query on the target ciphertext.

In the game, challenger $C$ with $\Im$ generates the following interaction procedure.

**Initial:** $C$ runs the *Setup* algorithm to generate the master key $s$ and the system parameters *params*. If the adversary is type $I$, $C$ sends *params* to $\Im_I$. Else, $C$ sends *params* and $s$ to $\Im_{II}$.

**Phase 1:** $\Im$ launches adaptively various oracles queries to $C$ but the rules of adversary behavior defined above are to be observed. $\Im_{II}$ cannot perform public key replacement query.

**Challenge:** $\Im$ sends two message $(m_0, m_1)$ of the same length, $n$ potential senders to form a public key ring $RL$, and the receiver's $ID_j$ to $C$. $C$ sends the ciphertext $\sigma$ to $\Im$.

**Phase 2:** Adversary $\Im$ performs adaptive queries, as in phase 1, but $\Im_I$ cannot perform a private key query on the target identity $ID_j$ and also cannot ask an unsigncryption query on the target ciphertext $\sigma^*$.

**Guess:** $\Im$ guesses $\mu^*$, if $\mu^* = \mu$, then game wins; otherwise, the game fails. $\Im$'s probability of winning this game is $Adv(\Im) = |2Pr[\mu^* = \mu] - 1|$.

**Definition 4.** The scheme is said to have existential unforgeability against adaptive chosen messages attack (EUF-CMA) if adversary $\Im$ does not have a non-negligible probability advantage in the unforgeability game.

In the game, adversary $\Im_I$ cannot query the private key and cannot request the private key if the corresponding public key has been replaced unless $\Im_I$ submits the corresponding secret value to $C$. $\Im_I$ cannot query both the public key, which replaced the target identity before the challenge phase and the signcryption.

Then, adversary $\Im_{II}$ cannot query the private key and replace the public key.

In the game, challenger $C$ with $\Im$ generates the following interaction procedure.

**Initial:** $C$ runs the *Setup* algorithm to generate the master key $s$ and the system parameters *params*. If the adversary is type $I$, $C$ sends *params* to $\Im_I$. Else, $C$ sends *params* and $s$ to $\Im_{II}$.

**Phase 1:** $\Im$ launches adaptively various oracles queries to $C$ as in the previous game.

**Forgery:** $\Im$ uses a public key ring $RL$, and the receiver's $ID_j$ to produce a ciphertext $\sigma^*$. If the forgery that satisfies the result of the unsigncryption query is not a $\perp$ and is not the result of a previous signcryption query. Also if the adversary belongs to type $I$, $\Im_I$ cannot query for both partial private and public key replacement. $\Im$ can win the game.

$\Im$'s probability of winning this game is $Adv(\Im) = |Pr[\Im\ wins]|$.

## 3 Certificateless Ring Signcryption Scheme

In this section, we show an efficient certificateless ring signcryption scheme with constant ciphertext length. For ease of illustration, we label the sender as $A$ and the receiver as $B$.

**Setup.** Input security parameter $k$, KGC selects an additive group $G_p$ of large prime order $p > 2^k$. Randomly choose the system master key $s \in Z_p^*$ and compute $P_{pub} = sP$ as the system public key, where $P$ is a generator of $G_p$. Define five hash functions $H_1 : \{0,1\}^* \rightarrow Z_p^*$, $H_2 : G_p \times G_p \rightarrow Z_p^*$, $H_3 : G_p \rightarrow Z_p^*$, $H_4 : G_p \rightarrow \{0,1\}^*$ and $H_5 : \{0,1\}^* \times G_p \times G_p \times G_p \times \{0,1\}^* \times G_p \times Z_p^* \rightarrow Z_p^*$.

KGC publishes parameters $params =< G_p, p, P, P_{pub}, H_1, H_2, H_3, H_4, H_5 >$ on blockchain and saves $s$ secretly.

**Set Partial Private Key.** The sender $A$ randomly chooses an integer $u_A \in Z_p^*$ as its secret value, calculates $U_A = u_A P$, and then sends $ID_A + u_A P_{pub}$ and $U_A$ to the KGC through the public channel. Because only KGC holds the system master key $s$, after receiving $ID_A + u_A P_{pub}$ and $U_A$ from the sender $A$, KGC can obtain the identity $ID_A$ of the sender $A$ in the form of $ID_A + u_A P_{pub} - sU_A = ID_A + su_A P - su_A P = ID_A$. Then, KGC randomly chooses an integer $y_A \in Z_p^*$ and computes the partial public key $Y_A = y_A P$ and partial private key $d_A = l_A y_A + s(mod\ p)$ of the sender $A$, where $l_A = H_1(ID_A)$.

KGC sends $d_A + sU_A$ and $Y_A + sU_A$ to the sender $A$ through the public channel.

**Set Public Key.** Since only the sender $A$ knows its secret value $u_A$, after receiving $d_A + sU_A$ and $Y_A + sU_A$ from KGC, the sender $A$ can obtain the private key $d_A$ and the partial public key $Y_A$ in the form of $d_A + sU_A - u_A P_{pub} =$

$d_A + su_AP - su_AP = d_A$ and $Y_A + sU_A - u_AP_{pub} = Y_A + su_AP - su_AP = Y_A$, respectively.

After that, the sender $A$ verifies whether the equation $d_AP = H_1(ID_A)Y_A + P_{pub}$ holds by **Equation** 1.

$$\begin{aligned} d_AP =& (l_Ay_A + s)P \\ =& l_AY_A + P_{pub} \\ =& H_1(ID_A)Y_A + P_{pub}. \end{aligned} \tag{1}$$

If it holds, the sender $A$ accepts $d_A$ and $Y_A$ and computes its own public key $PK_A = U_A + H_1(ID_A)Y_A$, after which $PK_A$ is published on blockchain. Otherwise, the sender $A$ rejects $d_A$ and $Y_A$.

**Set Private Key.** After the sender $A$ determines the validity of the partial private key $d_A$, the private key is set to $SK_A = (u_A, d_A)$ and stored secretly.

**Signcryption.** The sender $A$ randomly selects $n-1$ public keys of other users on blockchain to form the public key ring $RL = \{PK_1, PK_2, ..., PK_n\}$ for this signcryption and embed it in the sender of the transaction on blockchain, where the public key $PK_A \in RL$ of the sender $A$.

The sender $A$ randomly picks $r \in Z_p^*$ and computes $R = rP$, $Q = r(PK_B + P_{pub})$ and $q = H_2(Q, R)$.

Then randomly select $\theta \in Z_p^*$ and let $\alpha = \theta P$, construct $f(x) = (x - q)(\beta \sum_{i=1, i\neq A}^{n} PK_i + \theta PK_A - (\beta - \theta)P_{pub}) + \theta P = xM - qM + \alpha = xM + \vartheta$, where $\beta = H_3(\alpha)$, $M = \beta \sum_{i=1, i\neq A}^{n} PK_i + \theta PK_A - (\beta - \theta)P_{pub}$ and $\vartheta = -qM + \alpha$.

The sender $A$ makes $S = H_4(\vartheta)$, $c = S \oplus m$ and $V = \beta c$, computes $W = r^{-1}(u_A + d_A)(\beta - \theta)$ and $Z = H_5(S, \alpha, M, \vartheta, V, R, W)$. Generate the ciphertext $\sigma = \{M, \vartheta, R, V, W, Z, t_1\}$.

When posting the ciphertext $\sigma$ on blockchain via a transaction, the current timestamp $t_1$ needs to be added, where $t_i$ is the timestamp.

**Unsigncryption.** After the receiver $B$ gets the ciphertext $\sigma$, it first checks whether $t_2 - t_1$ is within the time threshold $\triangle t$, where $t_2$ is the current timestamp. If it is not satisfied, the failure symbol $\perp$ is output. Otherwise, continue to complete the unsigncryption process.

The receiver $B$ uses its own private key $SK_B$ to compute $Q' = (u_B + d_B)R$ and gets $q' = H_2(Q', R)$.

After that, construct $f(x) = xM + \vartheta$, bring in $q'$ to calculate $\alpha' = f(q')$ and get $\beta' = H_3(\alpha')$.

By computing $c' = \beta'^{-1}V$ and $S = H_4(\vartheta)$, the receiver $B$ gets the message $m' = S \oplus c'$.

The receiver $B$ needs to verify that $Z = H_5(S, \alpha', M, \vartheta, V, R, W)$ holds. If it does not hold, output the failure symbol $\perp$. Otherwise, continue to complete the unsigncryption process.

The receiver $B$ gets the public key ring $RL = \{PK_1, PK_2, ..., PK_n\}$ of this signcryption by looking at the sender of the transaction on blockchain. Check

whether the equation $WR = \beta \sum_{i=1}^{n} PK_i - M$ holds by **Eq.** 2.

$$\begin{aligned} WR =& r^{-1}(u_A + d_A)(\beta - \theta)rP \\ =& (\beta - \theta)(PK_A + P_{pub}) \\ =& (\beta - \theta)(\sum_{i=1}^{n} PK_i - \sum_{i=1, i\neq A}^{n} PK_i + P_{pub}) \\ =& \beta \sum_{i=1}^{n} PK_i - \beta \sum_{i=1, i\neq A}^{n} PK_i - \theta PK_A + (\beta - \theta)P_{pub} \\ =& \beta \sum_{i=1}^{n} PK_i - M. \end{aligned} \tag{2}$$

If it holds, it means that the message $m$ computed by the receiver $B$ is correct. Otherwise, the failure symbol $\perp$ is output. After passing the check, the receiver $B$ completes the unsigncryption process.

## 4 Security Analysis

**Theorem 1.** *Under the random oracle model, if there exists an adversary $\Im_I$ with probability advantage $\epsilon$ that wins the IND-CCA2-I security model game. Then $C$ can solve the CDHP with probability advantage $\frac{\epsilon}{q_{H_1} q_{H_2}}$.*

***Proof.*** In Game-I, challenger $C$ receives an instance $(P, aP, bP)$ of the CDHP and wants to solve the CDHP with $\Im_I$ and generates the following interaction procedure.

**Initial:** $C$ runs the *Setup* algorithm to generate the master key $s$ and the parameters *params*. After that $C$ sends *params* to $\Im_I$ and saves $s$ secretly.

**Phase 1:** $\Im_I$ launches adaptively various oracles queries to $C$. $C$ maintains corresponding lists, which are initially empty, and holds the queries and answers.

- $H_1$ queries: For the $(ID_i)$ queries, $C$ checks whether the list exists $(ID_i, l_i)$ and returns $l_i$ if it exists, otherwise it randomly selects $l_i \in Z_p^*$ and inserts it into the list $L_1$ and then returns $l_i$.
- $H_2$ queries: For the $(Q_i, R_i)$ queries, $C$ checks whether the list exists $(Q_i, R_i, q_i)$ and returns $q_i$ if it exists, otherwise it randomly selects $q_i \in Z_p^*$ and inserts it into the list $L_2$ and then returns $q_i$.
- $H_3$ queries: For the $(\alpha_i)$ queries, $C$ checks whether the list exists $(\alpha_i, \beta_i)$ and returns $\beta_i$ if it exists, otherwise it randomly selects $\beta_i \in Z_p^*$ and inserts it into the list $L_3$ and then returns $\beta_i$.
- $H_4$ queries: For the $(\vartheta_i)$ queries, $C$ checks whether the list exists $(\vartheta_i, S_i)$ and returns $S_i$ if it exists, otherwise it randomly selects $S_i \in Z_p^*$ and inserts it into the list $L_4$ and then returns $S_i$.
- $H_5$ queries: For the $(S_i, \alpha_i, M_i, \vartheta_i, V_i, R_i, W_i)$ queries, $C$ checks whether the list exists $(S_i, \alpha_i, M_i, \vartheta_i, V_i, R_i, W_i, Z_i)$ and returns $Z_i$ if it exists, otherwise it randomly selects $Z_i \in Z_p^*$ and inserts it into the list $L_5$ and then returns $Z_i$.

- Partial private key queries: $\Im_I$ sends the $(ID_i)$ to $C$, if the target identity $i = l$, $C$ fails and stops. Otherwise, $C$ checks whether the list exists $(ID_i, u_i, d_i, SK_i, PK_i)$ and returns $d_i$ if it exists, otherwise it randomly selects $d_i \in Z_p^*$ and inserts it into the list $L_k$, then $d_i$ is returned.
- Public key queries: $\Im_I$ sends the $(ID_i)$ to $C$, $C$ checks whether the list exists $(ID_i, u_i, d_i, SK_i, PK_i)$ and returns $PK_i$ if it exists, otherwise it randomly selects $u_i, d_i, l_i \in Z_p^*$, sets $l_i = H_1(ID_i)$ and $Y_i = (d_i P - P_{pub}) l_i^{-1}$, and computes $PK_i = U_i + l_i Y_i$, where $U_i = u_i P$ and $d_i P = l_i Y_i + P_{pub}$. Then $C$ inserts it into the list $L_k$, then $PK_i$ is returned.
- Private key queries: $\Im_I$ sends the $(ID_i)$ to $C$, if the target identity $i = l$, $C$ fails and stops. Otherwise, $C$ checks whether the list exists $(ID_i, u_i, d_i, SK_i, PK_i)$ and returns $d_i$ if it exists, otherwise it randomly selects $u_i, d_i \in Z_p^*$ and inserts it into the list $L_k$, then $SK_i$ is returned.
- Public key replacement queries: $\Im_I$ sends the replaced public key $(ID_i, PK_i')$ to $C$, $C$ updates the list $L_k$ and reset the information $(ID_i, \perp, \perp, \perp, PK_i')$.
- Signcryption queries: $\Im_I$ sends the message $m$, $n$ potential senders to form a public key ring $RL$, and the receiver's $ID_j$ to $C$. If the target identity $i \neq l$, then $C$ knows the private key $SK_i$ of the sender and returns the result according to the *Signcryption* algorithm. Otherwise, $C$ randomly selects $r \in Z_p^*$ and computes $R_i = rP$. Find $(Q_i, R_i, q_i)$ from the list $L_2$, randomly select $\theta \in Z_p^*$ and compute $f(x) = (x - q_i)(\beta_i \sum_{i=1, i \neq A}^{n} PK_i + \theta PK_A - (\beta_i - \theta) P_{pub}) + \theta P = xM_i + \vartheta_i$. Find $(\alpha_i, \beta_i)$ and $(\vartheta_i, S_i)$ from the list $L_3$ and $L_4$, respectively. $C$ computes $c_i = S_i \oplus m$ and $V_i = \beta_i c_i$. Set $k_j = u_j + d_j$ and compute $W_i = r^{-1} k_j (\beta_i - \theta)$. Find $(S_i, \alpha_i, M_i, \vartheta_i, V_i, R_i, W_i, Z_i)$ from the list $L_5$ and send the ciphertext $\sigma = \{M, \vartheta, R, V, W, Z\}$ to $\Im_I$.
- Unsigncryption queries: $\Im_I$ sends the ciphertext $\sigma = \{M, \vartheta, R, V, W, Z\}$, and the receiver's $ID_j$ to $C$. If the target identity $j \neq l$, then $C$ knows the private key $SK_j$ of the receiver and returns the result according to the *Unsigncryption* algorithm. Otherwise, $C$ rejects the ciphertext $\sigma$.

**Challenge:** $\Im_I$ sends two message $(m_0, m_1)$ of the same length, $n$ potential senders to form a public key ring $RL$, and the receiver's $ID_j$ to $C$. If the target identity $ID_l \neq ID_j$, then $C$ fails. Otherwise, $C$ randomly selects $\mu \in \{0, 1\}$, computes $Q_j = k_j R$. Randomly select $q, \theta \in Z_p^*$ and compute $f(x) = (x - q)(\beta \sum_{i=1, i \neq A}^{n} PK_i + \theta PK_A - (\beta - \theta) P_{pub}) + \theta P = xM + \vartheta$. Then, $C$ selects $W \in Z_p^*$, sends the ciphertext $\sigma = \{M, \vartheta, R = rP, V_\mu = \beta c_\mu, W, Z = H_5(S, \alpha, M, \vartheta, V_\mu, R, W)\}$ to $\Im_I$, where $\beta = H_3(\alpha)$, $\vartheta = -qM + \alpha$, and $S = H_4(\vartheta)$.

**Phase 2:** Adversary $\Im_I$ performs adaptive queries, as in phase 1, but $\Im_I$ cannot perform a private key query on the target identity $ID_j$ and also cannot ask an unsigncryption query on the target ciphertext $\sigma^*$.

**Guess:** $\Im_I$ submits the $\mu^*$ to determine whether $\mu^* = \mu$ holds.

However, the target ciphertext $\sigma^*$ given to $\Im_I$ is randomly distributed in the ciphertext space and $\Im_I$ cannot gain any advantage in this simulation. we know $C$ selects a random $(Q, R, q)$ from the list $L_2$ and takes the corresponding $R$ and $Q$ as the solution to solve the CDHP. Therefore, any adversary with advantage $\epsilon$

in the real IND-CCA2-I game must be aware with probability at least $\epsilon$ that the challenge ciphertext provided by $C$ is wrong. In order for the $\Im_I$ to discover that $\sigma$ is not a valid ciphertext, the $\Im_I$ should query the $q_{H_2}$ oracle with $Q_j = k_j R_j$. Here, $k_j$ is the private key of the target identity, which is $PK_j + P_{pub} = aP$. In addition, $C$ sets $R^* = bP$. As a result, $Q_j = k_j R_j^* = abP$. Therefore, one of the entries from the list $L_2$ should be the value $abP$. $C$ selects the value $Q_j$ from the list $L_2$ with probability $\frac{1}{q_{H_1}}$, which would be the solution for CDHP.

Under the IND-CCA2-I security model game, the probability advantage about the event that $C$ successfully solves the CDHP is $\frac{\epsilon}{q_{H_1} q_{H_2}}$.

**Theorem 2.** *Under the random oracle model, if there exists an adversary $\Im_{II}$ with probability advantage $\epsilon$ that wins the IND-CCA2-II security model game. Then $C$ can solve the CDHP with probability advantage $\frac{\epsilon}{q_{H_1} q_{H_2}}$.*

***Proof.*** In Game-II, challenger $C$ receives an instance $(P, aP, bP)$ of the CDHP and wants to solve the CDHP with $\Im_{II}$. $C$ runs the *Setup* algorithm to generate the master key $s$ and the system parameters *params*. After that $C$ sends *params* and $s$ to $\Im_{II}$. The proof of Game-II is very similar to the proof of Game-I, except for public key replacement queries. $C$ selects the value $Q_j$ from the list $L_2$ with probability $\frac{1}{q_{H_1}}$, which would be the solution for CDHP.

Under the IND-CCA2-II security model game, the probability advantage about the event that $C$ successfully solves the CDHP is $\frac{\epsilon}{q_{H_1} q_{H_2}}$.

**Theorem 3.** *Under the random oracle model, if there exists an adversary $\Im_I$ with probability advantage $\epsilon$ that wins the EUF-CMA-I security model game. Then $C$ can solve the DLP with probability advantage $\frac{\epsilon^2}{66 C_{q_{H_3}}^n} \frac{1}{q_{H_1} q_{H_5}}$.*

***Proof.*** In Game-III, challenger $C$ receives an instance $(P, aP)$ of the DLP and wants to solve the DLP with $\Im_I$ and generates the following interaction procedure.

Initial and Phase 1 of Game-III are similar in Game-I.

**Forgery:** $\Im_I$ forges the ciphertext $\sigma = \{M, \vartheta, R, V, W, Z\}$ and submits it with the help of the message $m$ and the public key ring $RL$.

By the forking lemma [4], $\Im_I$ will output four additional signncryptions after using the same random tape replay but selecting different $H_1$, $H_2$, $H_3$, $H_4$ and $H_5$ with probability $\frac{\epsilon^2}{66 C_{q_{H_3}}^n}$.

we know $C$ selects a random $(S, \alpha, M, \vartheta, V, R, W, Z)$ from the list $L_5$ and takes the corresponding $R$ and $W$ as the solution to solve the DLP. In order for the $\Im_I$ to discover that $\sigma$ is not a valid ciphertext, the $\Im_I$ should query the $q_{H_5}$ oracle with $WR = (\beta - \theta)(PK_A + P_{pub})$. Here, $u_i$ is the secret value of the target identity, which is $U_i = aP$. As a result, $a = WR(\beta P - \alpha)^{-1} - d_i$. Therefore, one of the entries from the list $L_5$ should be the value $a$. $C$ selects the value $W$ from the list $L_5$ with probability $\frac{1}{q_{H_1}}$, which would be the solution for DLP.

Under the EUF-CMA-I security model game, the probability advantage about the event that $C$ successfully solves the DLP is $\frac{\epsilon^2}{66 C_{q_{H_3}}^n} \frac{1}{q_{H_1} q_{H_5}}$.

**Theorem 4.** *Under the random oracle model, if there exists an adversary* $\Im_{II}$ *with probability advantage* $\epsilon$ *that wins the EUF-CMA-II security model game. Then* $C$ *can solve the DLP with probability advantage* $\frac{\epsilon^2}{66C^n_{q_{H_3}}} \frac{1}{q_{H_1} q_{H_5}}$.

***Proof.*** In Game-IV, challenger $C$ receives an instance $(P, aP)$ of the DLP and wants to solve the DLP with $\Im_{II}$. The proof of Game-IV is very similar to the previous proof, except for public key replacement queries. $C$ selects the value $W$ from the list $L_5$ with probability $\frac{1}{q_{H_1}}$, which would be the solution for DLP.

Under the EUF-CMA-I security model game, the probability advantage about the event that $C$ successfully solves the DLP is $\frac{\epsilon^2}{66C^n_{q_{H_3}}} \frac{1}{q_{H_1} q_{H_5}}$.

**Anonymity.** In the certificateless ring signcryption scheme of this paper, it is easy to understand that this scheme is unconditionally anonymous. All parameters mentioned are independent and uniformly distributed for any message $m$, receiver, and public key ring $RL$, regardless of who is the actual signer. Even an adversary with all private keys corresponding to the public key ring $RL$ and infinite computational resources cannot identify the real signcrypter with better probability than a random guess, and that guess will be equal to $\frac{1}{n}$, where $n$ denotes the number of public keys selected in the signcryption process. Therefore, this scheme has anonymity.

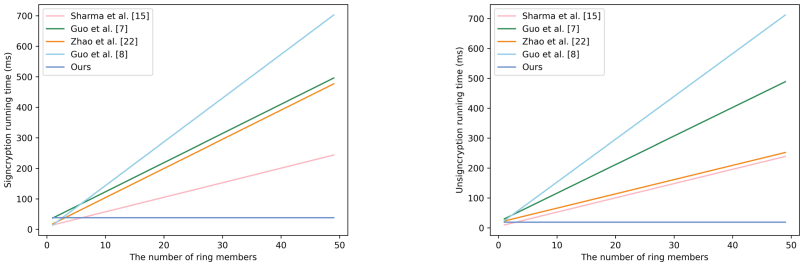
## 5 Performance Analysis

In this paper, we compare the performance of the certificateless ring signcryption scheme with Sharma et al. [15], Guo et al. [7], Zhao et al. [22] and Guo et al. [8].

**Table 1.** Performance comparison of certificateless ring signcryption schemes

| Schemes | Signcryption | Unsigncryption | Ciphertext Size |
|---|---|---|---|
| Sharma et al. [15] | $(n+2)PM+(2n+1)H$ | $(n+1)PM+(n+2)H$ | $2\|m\|+(n+2)\|G_1\|$ |
| Guo et al. [7] | $1P+(2n+5)PM+(n+2)H$ | $3P+(2n+2)PM+(n+2)H$ | $(n+1)\left\|Z_p^*\right\|+1\|m\|+(n+3)\|G_1\|$ |
| Zhao et al. [22] | $1P+(2n+1)PM+(n+1)H$ | $5P+(n)PM+(n+1)H$ | $1\|m\|+(3n+3)\|G_1\|$ |
| Guo et al. [8] | $(3n)PM+(n+3)H$ | $(3n+2)PM+(n+2)H$ | $(n+1)\left\|Z_p^*\right\|+1\|m\|+2\|G_1\|$ |
| Ours | $8PM+4H$ | $4PM+4H$ | $2\left\|Z_p^*\right\|+1\|m\|+3\|G_1\|$ |

The experiments in this paper are based on Intel(R) Core(TM) i5-11400 CPU @ 2.60GHz, 16.00GB RAM and PBC library [11] on Windows 10. Ignoring scalar multiplication operation in $Z_p^*$ and addition operation, the results are shown in Table 1. Where $H$ denotes the hash function operation, $PM$ denotes the scalar point multiplication operation in $G_1$, and $P$ denotes the bilinear pairing operation. $|m|$ denotes the length of the message, $\left|Z_p^*\right|$ denotes the length of the element in $Z_p^*$, and $|G_1|$ denotes the length of the element in $G_1$.




(a) Signcryption cost (b) Unsigncryption cost

**Fig. 1.** Computation cost

To show the performance difference between the schemes more intuitively, we can show the results in Fig. 1. It can be seen that our scheme has less computation cost than the other schemes in the signcryption process and the unsigncryption process, the computation overhead does not vary with the size of the ring both in Fig. 1a and Fig. 1b. Also, we can see that the ciphertext length used in our scheme is constant, which is more advantageous than the above schemes. Therefore, the performance of our scheme in this paper is better than the above schemes.

## 6 Conclusion

In this paper, we propose a new certificateless ring signcryption scheme with constant ciphertext length on blockchain, which optimizes the computation cost and ciphertext length compared to other schemes and proves the security of this scheme under the random oracle model. The scheme outperforms other schemes in terms of computational overhead and ciphertext length and does not vary with the ring size. In the future, continuing the optimization of the scheme and extending it to electronic voting systems on blockchain will be the next research direction.

## References

1. Abdeldaym, R.S., Abd Elkader, H.M., Hussein, R.: Modified RSA algorithm using two public key and Chinese remainder theorem. IJ Electron. Inf. Eng. **10**(1), 51–64 (2019)
2. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40061-5_29
3. Chandran, N., Groth, J., Sahai, A.: Ring signatures of sub-linear size without random oracles. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 423–434. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73420-8_38
4. Deng, L., Liu, C., Wang, X.: An improved identity-based ring signcryption scheme. Inf. Secur. J.: Glob. Perspect. **22**(1), 46–54 (2013)

5. Dodis, Y., Kiayias, A., Nicolosi, A., Shoup, V.: Anonymous identification in *Ad Hoc* groups. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 609–626. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_36
6. Fujisaki, E., Suzuki, K.: Traceable ring signature. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 181–200. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71677-8_13
7. Guo, H., Deng, L.: Certificateless ring signcryption scheme from pairings. Int. J. Netw. Secur. **22**(1), 102–111 (2020)
8. Guo, R., Xu, L., Li, X., Zhang, Y., Li, X.: An efficient certificateless ring signcryption scheme with conditional privacy-preserving in VANETs. J. Syst. Architect. **129**, 102633 (2022)
9. Jothi, A.A., Srinivasan, B.: Security analysis in body area networks using attribute-based ring signcryption scheme. Res. J. Appl. Sci. Eng. Technol. **13**(1), 48–56 (2016)
10. Ta, A.T., et al.: Efficient unique ring signature for blockchain privacy protection. In: Baek, J., Ruj, S. (eds.) ACISP 2021. LNCS, vol. 13083, pp. 391–407. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90567-5_20
11. Lynn, B.: PBC library: the pairing-based cryptography library, 2013. https://crypto.stanford.edu/pbc/. Accessed 1 May 2022
12. Noether, S., Mackenzie, A., et al.: Ring confidential transactions. Ledger **1**, 1–18 (2016)
13. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 552–565. Springer, Berlin, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_32
14. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Workshop on the Theory and Application of Cryptographic Techniques, pp. 47–53. Springer, Berlin, Heidelberg (1984). https://doi.org/10.1007/3-540-39568-7_5
15. Sharma, G., Bala, S., Verma, A.K.: Pairing-free certificateless ring signcryption (PF-CLRSC) scheme for wireless sensor networks. Wireless Pers. Commun. **84**(2), 1469–1485 (2015). https://doi.org/10.1007/s11277-015-2698-2
16. Shen, H., Chen, J., He, D., Shen, J.: Insecurity of a pairing-free certificateless ring signcryption scheme. Wireless Pers. Commun. **96**(4), 5635–5641 (2017)
17. Sun, S.-F., Au, M.H., Liu, J.K., Yuen, T.H.: RingCT 2.0: a compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) ESORICS 2017. LNCS, vol. 10493, pp. 456–474. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66399-9_25
18. Wang, L., Zhang, G., Ma, C.: A secure ring signcryption scheme for private and anonymous communication. In: 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), pp. 107–111. IEEE (2007)
19. Yu, H., Zhang, S., Liu, Y.: Certificateless multivariate ring signcryption scheme. J. Inf. Secur. Appl. **68**, 103220 (2022)
20. Yuan, Y., Wang, F.: Current status and prospects of blockchain technology development. Acta Autom. Sinica **42**(4), 481–494 (2016)
21. Zhang, S., Zhao, Y., Wang, B.: Certificateless ring signcryption scheme for preserving user privacy in smart grid. Autom. Electr. Power Syst. **42**(3), 23–118 (2018)
22. Zhao, N., Zhang, G.: Authenticated privacy protection scheme based on certificateless ring signcryption in VANET. Comput. Sci. **47**(3), 312–319 (2020)

# An Efficient Soft Analytical Side-Channel Attack on Ascon

Sinian Luo[1], Weibin Wu[1], Yanbin Li[2], Ruyun Zhang[3], and Zhe Liu[1(✉)]

[1] Nanjing University of Aeronautics and Astronautics, Nanjing, China
{luosinian,wuweibin,zhe.liu}@nuaa.edu.cn
[2] College of Artificial Intelligence, Nanjing Agricultural University, Nanjing, China
yanbinli@njau.edu.cn
[3] Zhejiang Lab, Hangzhou, China
zhangry@zhejianglab.com

**Abstract.** Lightweight cryptography is a subfield of cryptography, which is widely used in embedded systems, RFID, sensor networks, and so on. However, the leakage information during the operation of these IoT devices can be exploited by adversaries and subjected to side-channel attacks. Simultaneously, only a small number of previous works show these attacks. In this work, we perform the soft analytical side-channel attack (SASCA) on the encryption of Ascon. Since we construct a unique factor graph for Ascon, we can also use it to attack the masked implementations. The point of attack is the permutation function, one of Ascon's most basic components. Our attack mainly consists of three steps. At the first, we run a side-channel template matching on the initialization phase. Then, we build a factor graph describing the intermediate computations in permutation, including the observed leakage for the intermediate variables. Third, we run a Belief Propagation (BP) algorithm that takes full advantage of these leakages. Through simulations, we show that the entire key can be successfully recovered by only using the leakage information of a few traces, and it also offers low time and memory complexity.

**Keywords:** Lightweight cryptography · Side-channel attacks · Belief propagation

## 1 Introduction

The security of IoT devices in constrained environments is one of the main challenges faced by industrial control systems. Therefore, various universities and scientific research institutions have carried out research on the design of lightweight cryptography over the past decade. In March 2021, the National Institute of Standards and Technology (NIST) announced the ten finalists for the last round of the Lightweight Crypto (LWC) Competition. It is worth noting that Ascon, designed by Dobraunig et al. [3], was selected as one of the finalists since its high performance and easy to implement. Ascon has been selected as


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 389–400, 2022.
https://doi.org/10.1007/978-3-031-19208-1_32

the primary choice for lightweight authenticated encryption in the final portfolio of the Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR).

After authentication encryption Ascon was proposed, there has been a series of security analysis works. In 2016, Samwel [13] successfully implemented the Differential Power Analysis on an unprotected implementation in FPGA but failed on an implementation protected with a threshold implementation [9]. Gross et al. [5] proposed implementations of Ascon suitable for IoT devices. Besides, they provided protected implementations of Ascon for applications that could be resistant to first-order DPA attacks. Samwel et al. [14] presented and applied attacks on simple hardware implementations. Then they used a third-order distinguisher to attack a toy version of Ascon that was protected by a threshold implementation. In 2018, Adomnicai et al. [1] proposed a masking scheme for Ascon, which can protect not only the initialization phase but also the finalization phase.

The strategy of side-channel attacks can be classified as Divide-and-Conquer (DC) and analytical. For DC attacks, they attack individual parts of the algorithm and then analyze the information they get. These attacks are simple to use and have high applicability, but they have high data complexity and time complexity, and they are easy to protect. By contrast, the analytical strategy focuses more on using mathematical analysis to recover keys after obtaining physical leaks. Thus, these attacks can succeed when the adversary has very limited available information (vs. DC attacks). The Algebraic Side-Channel Attacks (ASCA) proposed by Renauld et al. [12] is a typical representative of the analysis strategy. Subsequently, Veyrat-Charvillon et al. [15] proposed SASCA, which combines the advantages of the divide and conquer strategy and ASCA. SASCA not only has the simplicity and versatility of DC attacks but also uses mathematical analysis to make full use of leakage.

In this work, according to the encryption of Ascon, we construct a factor graph for the initialization phase that introduces the entire key. Then run SASCA on the factor graph. We show that few traces attacks are indeed a threat to implementations of lightweight cryptography. Furthermore, we simplify the whole factor graph, extract the crucial operations and remove the irrelevant parts. We present an efficient SASCA attack based on the simplified factor graph that recovers the entire key with only a few encrypted side-channel observations and adequate leakage. Since our attack is only performed in the initialization phase, the masking scheme proposed by Adomnicai et al. [1] is theoretically unprotected.

**Outline.** In Sect. 2, we first give an introduction to our attack target on Ascon and the BP algorithm, which we use as a basis for our approach. Our side-channel attack is then introduced in Sect. 3. We recall the main steps of the attack and then made optimizations for SASCA, which dramatically enhanced the performance of the attack. In Sect. 4, we show the consequence of our attack

using simulations. Finally, we summarize the performance of our attack and explore some open questions in Sect. 5.

## 2 Background

In this section, we brifiey recall our attack target, namely Ascon, as well as the BP algorithm, which is the crucial process of SASCA.

### 2.1 Ascon

Ascon authenticated encryption is based on a duplex construction. It has two different versions, Ascon-128 and Ascon-128a, with 64 and 128 bit block size and parameters as shown in Table 1.

**Table 1.** Parameters for Ascon authenticated encryption

| Name | Bit size of | | | | | Rounds | |
|---|---|---|---|---|---|---|---|
| | key | nonce | tag | data block | capacity | $p^a$ | $p^b$ |
| Ascon-128 | 128 | 128 | 128 | 64 | 256 | 12 | 6 |
| Ascon-128a | 128 | 128 | 128 | 128 | 192 | 12 | 8 |

All Ascon versions use the same permutation $p$ which operates on a state size of 320-bit (consists of 5 words $x_0, x_1, x_2, x_3, x_4$, each 64-bit). So our attack in this paper is targeted at Ascon-128. As shown in Fig. 1, the encryption process of Ascon is split into four phases: initialization, processing associated data, processing the plaintext, and finalization. These phases update with permutation $p$ which circularly runs round transformation $a$ rounds or $b$ rounds. The round transformation mainly includes the following three processes.

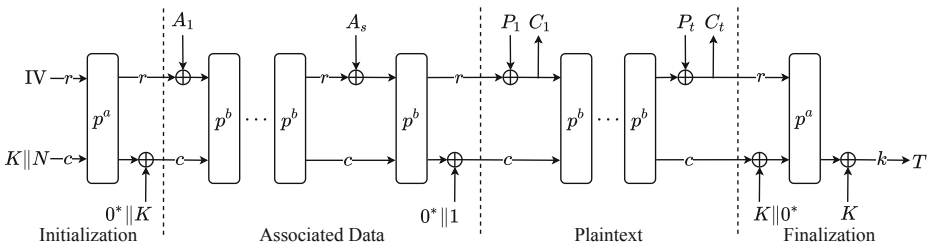
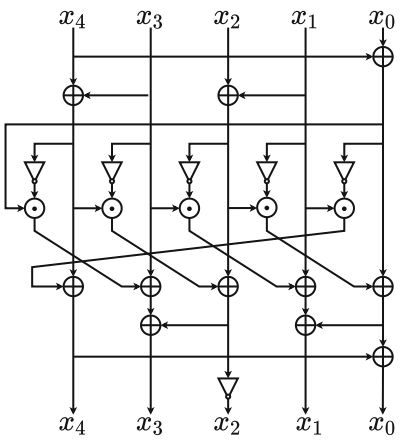



**Fig. 1.** Encryption and authentication of Ascon

**Addition of Constants:** Adds a specific constant to word $x_2$ of the state $S$.

**Nonlinear Substitution Layer:** Applies a 5-bit S-box 64 times to each bit-slice of $S$ in parallel. Ascon's S-box implementation is almost similar to the Keccak [2] $\chi$ mapping. In addition, the S-box can be implemented with some logical operations.(see Fig. 2 left part).

**Linear Diffusion Layer:** The five registers words is rotated twice with different rotation values and XOR to itself.(see Fig. 2 right part).




$$x_0 \leftarrow x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28)$$
$$x_1 \leftarrow x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39)$$
$$x_2 \leftarrow x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6)$$
$$x_3 \leftarrow x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17)$$
$$x_4 \leftarrow x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)$$

**Fig. 2.** Nonlinear substitution layer (left) and linear diffusion layer (right)

### 2.2 Belief Propagation

The first BP algorithm was proposed in 1982 [10] and is usually used in Bayes nets and factor graphs. Our introduction of the BP algorithm mainly comes from MacKay [7, chapter 26]. Let us consider a function $P^*$ of a set of $N$ variables $\mathbf{x} \equiv \{x_n\}_{n=1}^{N}$ is defined as a product of $M$ factors as follows:

$$P^*(\mathbf{x}) = \prod_{m=1}^{M} f_m(\mathbf{x}_m),$$

where each of the factors $f_m(\mathbf{x}_m)$ is a function of a subset $\mathbf{x}_m$ of variables that make up $\mathbf{x}$. The problem of marginalization can be solved by the marginal function of any variable $x_n$ , defined by

$$Z_n(x_n) = \sum_{\{x_{n'}\}, n' \neq n} P^*(\mathbf{x}),$$

or its normalized marginal $P_n(x_n) = Z_n(x_n)/Z$, where:

$$Z = \sum_{\mathbf{x}} \prod_{m=1}^{M} f_m(\mathbf{x}_m).$$

All these solutions are intractable in general. There is an exponential relationship between the consumption of marginalization and the number of variables $N$.

The BP algorithm can be well illustrated by the message-passing principle. It expresses the desired relationship in terms of variables and factors into a bipartite factor graph. We use $\mathcal{N}(m)$ to denote the set of variables involved in factor $f_m$, $\mathcal{M}(n)$ to denote the set of factors in which variable $n$ participates. In addition, we use $\mathbf{x}_m \backslash n$ to denote the exclusion of $x_n$ from the set of variables in $\mathbf{x}_m$. The BP algorithmic message passing consists of two parts: from variable nodes to factor nodes ($q_{n\to m}$), and from factor nodes to variable nodes ($r_{m\to n}$). The update rules follow the following two rules:

$$q_{n\to m}(x_n) = \prod_{m' \in \mathcal{M}(n)\backslash m} r_{m'\to n}(x_n).$$

$$r_{m\to n}(x_n) = \sum_{\mathbf{x}_m \backslash n} \left( f_m(\mathbf{x}_m) \prod_{n' \in \mathcal{N}(m)\backslash n} q_{n'\to m}(x'_n) \right).$$

According to the above description, the tree-shaped factor graph can always converge with a finite number of iterations. After convergence, the marginal function (belief ) $Z_n(x_n)$ can be obtained by multiplying all the received information on the corresponding node:

$$Z_n(x_n) = \prod_{m \in \mathcal{M}(n)} r_{m\to n}(x_n),$$

and the normalized marginals $P_n(x_n) = Z_n(x_n)/Z$ can be calculated by $Z = \sum_{x_n} Z_n(x_n)$. If that is not the tree-shaped graphical model, convergence is not guaranteed, but the same update rules can still be used and often gives sufficiently accurate approximations to the real marginals (so-called "loop BP"). According to the characteristics of the BP algorithm, we can know that the time and memory complexity are mainly determined by the number of possible values for each variable and the number of edges, which will provide the theoretical basis for our subsequent optimization.

## 3 The Proposed Method

In this section, we detailedly describe the main attack method called SASCA, proposed by Veyrat-Charvillon et al. [15]. Then propose the appropriate factor graph for Ascon to improve the performance of the attack.

### 3.1 SASCA

The advantages and disadvantages of DC attacks such as DPA and ASCA are generally extreme. Although DPA has low time and memory complexity as well as high noise tolerance, it also has high data complexity (often requiring millions of traces). On the contrary, ASCA performs better at data complexity but is very sensitive to noise. SASCA just balances the characteristics between them and can tolerate more noise while having low data complexity in a way similar to a Low-Density Parity Check code (LDPC) [4]. In simple terms, SASCA first performs a template attack on the targeted variable and then obtains the posterior probabilities based on leakage. That is, for each intermediate $T$, we can obtain $Pr(T = t \mid L_{\text{out}})$ by template matching, where $L_{\text{out}}$ denotes the obtained leakage information and $t$ runs through all realizations of the intermediate $T$. More precisely, they can be illustrated by the following three processes:

1. *Construction.* The interesting part of the cryptographic algorithm is represented as a factor graph with two types of nodes and bidirectional edges. The intermediate value of the calculation is represented by the variable node. The function nodes consist of posterior probabilities obtained by template matching and the algorithm description for joining different variables. These nodes are linked by bidirectional edges and pass information to each other (for more information about factor graph, see [7]).
2. *Information extraction.* In the side-channel attack, we extract information about the variable nodes of interest in the previous step by using leakage trace in the cryptographic process. The posterior distribution of each target variable can be obtained by subsequently comparing the leaked information with the previously established templates.
3. *Decoding.* As same as LDPC codes, decoding is done via the BP algorithm (another way is the sum-product algorithm). It iterates the previously obtained information and finally gets the marginal probabilities of the target variable.

### 3.2 Basic Construction

As seen in Sect. 2.1, All Ascon versions use the same permutation $p$. The encryption and decryption can be viewed as multiple $p$ permutations stacked together. As shown in Fig. 3, we build the factor graph for the round of $p$ permutation (for simplicity, each variable node in the graph is not added with a function node containing posterior probabilities). It is important to note that, unlike LDPC codes, SASCA's function nodes are no longer a single OR but an arbitrary function in the cryptography algorithm. Therefore, there are five different function nodes in the graph. The first is the posterior probabilities of variables obtained through side-channel leakage, denoted as $f_i(T_i) = Pr(T_i = t \mid L_{\text{out}})$. Then, the XOR function xor the inputs of two variables in bitwise and outputs to a new variable. Next, the Sbox function is the same as Keccak's Sbox, which takes three variables as input and creates a new variable as the output. The NOT function is simply a bitwise inversion of the input variable. Finally, the ROR function

here does not mean a right-rotation (circular shift) but refers to all operations of the entire linear diffusion layer, as shown in Fig. 2. Table 2 shows all the function nodes used in the Ascon factor graph.

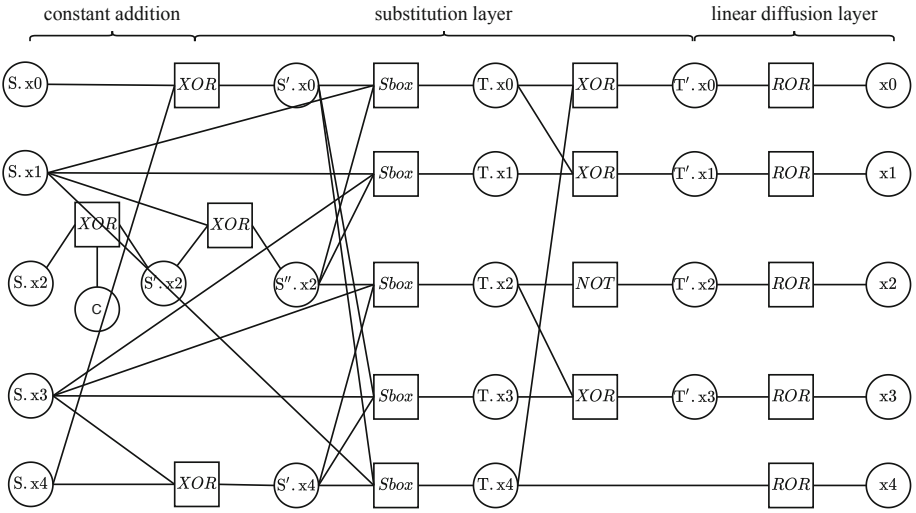



**Fig. 3.** Graph representation of the $p$ permutation round

**Table 2.** Summary of the function nodes

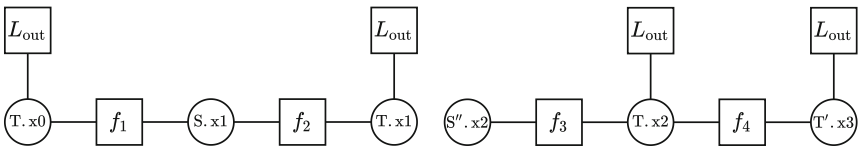
| $f_i(T_i) = Pr(T_i = t \mid L_{\text{out}})$ | |
|---|---|
| $XOR(A,B,C) = \begin{cases} 1 \text{ if} A \oplus B = C, \\ 0 \text{ otherwise.} \end{cases}$ | $Sbox(A,B,C,D) = \begin{cases} 1 \text{ if} A = Sbox(B,C,D), \\ 0 \text{ otherwise.} \end{cases}$ |
| $NOT(A,B) = \begin{cases} 1 \text{ if} A = NOT(B), \\ 0 \text{ otherwise.} \end{cases}$ | $ROR(A,B) = \begin{cases} 1 \text{ if} A = ROR(B), \\ 0 \text{ otherwise.} \end{cases}$ |

### 3.3 Optimize

Although the factor graph described above is simple in description and has low memory complexity, it has two significant drawbacks. Concretely, the factor graph represents all intermediate values in the rounds of the $p$ permutation as variable nodes, so the factor graph is no longer tree-shaped but contains cycles. Although we have described in Sect. 2.2, by a finite number of iterations, it is possible to make the BP converge and obtain a sufficiently accurate approximation of the real marginals. Factor graphs with loops introduce positive feedback, which can lead to overconfidence in certain beliefs. This means that when there

is a lot of noise in these beliefs, there will be more negative effects on the final result, followed by even oscillations, especially when deterministic factors are involved.

Therefore, in a practical side-channel attack, a perfect factor graph should avoid loops as much as possible. In addition, we found an interesting point in the actual attack. That is, in the latter part of the round of the $p$ permutation (such as the linear diffusion layer), the previous input part is mainly confused and diffused. In other words, when the intermediate calculation contained in the factor graph is effective enough, we can appropriately reduce some intermediate processes, which will not affect our recovery of the complete key.




(a) The factor graph used to recover $x_1$. (b) The factor graph used to recover $x_2$.

**Fig. 4.** Simplified factor graph

To alleviate these shortcomings, we analyze the 320-bit initial state $S$ (including 64-bit of IV specifying the algorithm, 128-bit of key $K$, as well as 128-bit of nonce $N$) of $p$ permutation for the entire initialization. We remove the linear layer and construct a simplified factor graph for the S-box part only, which, of course, does not include rings. Finally, the simplified factor graph used to recover key $x_1$ is shown in Fig. 4a, and the simplified factor graph used to recover key $x_2$ is shown in Fig. 4b.

In simplified factor graph, we treat operations on known values and variables as a one-to-one mapping of inputs and outputs, thus reducing some intermediate variables. For instance, the simplified S-box function can be described by factor $f_3$. When denoting $S''.x2$ as the input and $T.x2$ as the output, then $f_3$ is given as:

$$f_3(S''.x2, T.x2) = \begin{cases} 1 & \text{if } T.x2 = S''.x2 \oplus (\neg S.x3 \wedge S.x4), \\ 0 & \text{otherwise.} \end{cases}$$

The factor $f_4$ can be described as XOR operation of a given value and input variable:

$$f_4(T.x2, T'.x3) = \begin{cases} 1 & \text{if } T'.x3 = T.x2 \oplus T.x3, \\ 0 & \text{otherwise.} \end{cases}$$

Before optimization, 24 variable nodes and 41 function nodes were needed in a round, but after simplification, only 6 variable nodes and 7 function nodes were needed, reducing the amount by 3 times.

## 4 Experimental Results

Previously, SASCA first performs a template attack on the targeted intermediates and then incorporates leakage information into the factor graph. For this purpose, we aim at software (microcontroller) implementations of Ascon, such as the GitHub repositories with implementations (reference/optimized) [8], to specify the location of the leakages.

**Simulations with Leakage.** In order to allow for numerous repeated experiments and the reproducibility of our experiments, we evaluate our attack using leakage simulations. For all our simulations, we chose a more generic and simpler model called hamming weight with an additive Gaussian noise leakage model. In other words, the simulated leakage $L_{\text{out}}$ of intermediate value $x$ is given as:

$$L_{\text{out}} = \text{HW}(x) + \mathcal{N}(0, \sigma_{\text{HW}}),$$

where HW denotes the function of hamming weight and $\mathcal{N}$ the Gaussian distribution with zero mean and standard deviation $\sigma_{\text{HW}}$.

**Attack Implementation.** We implemented the method described in the previous section, including the template attack and the BP algorithm. Firstly, we perform a template matching on generated samples in C code and obtain the conditional probability of intermediate variables $\Pr(X = x \mid L_{\text{out}})$. For the most important part, the BP algorithm, we outsourced to Python code. All experiments were run on an Intel Core i7-8700 (3.20GHz).

The attack performance is evaluated for various values of $\sigma_{\text{HW}}$ parameters. We run at least 100 experiments for each situation and noise level under consideration. The success rate is calculated by counting the number of times the BP algorithm assigns the highest probability to the correct value in each of the 100 experiments.

The factor graph in our original attack was a common $p$ permutation for the entire cryptographic algorithm. Furthermore, because the latter part of $p$ permutation includes all shift operations (the linear diffusion layer) that do not modify the hamming weight of the variable, the hamming weight model we chose will no longer be valid. Simultaneously, it is proved through experiments that some weakly correlated operations are added to the factor graph (that is, they are not beneficial to spreading correct beliefs).

As a result, the factor graph of the original attack performs poorly in terms of efficiency and consistency. Subsequently, we run simulations in the 8-bit scenario for the optimized experiments. The results of the experiment are shown in Fig. 5. The X-axis represents the attack under different noise levels, and the Y-axis corresponds to the success rate of different noise levels. In this case, the attack can deal with up to $\sigma_{\text{HW}} = 1.2$ while maintaining a perfect success rate. As the noise increases, the success rate steadily drops to zero when $\sigma_{\text{HW}} = 4$. The whole forward-backward loop of BP in optimized attacks takes about 15 s on a

single core of our system, which is 4 times faster than the unoptimized attack. This result is better than SASCA, used in other cryptographic algorithms, such as AES [15], kyber [11] and Keccak [6]. This makes it possible to tolerate more noise in practical attacks, which greatly increases the threat of side-channel information leakage.

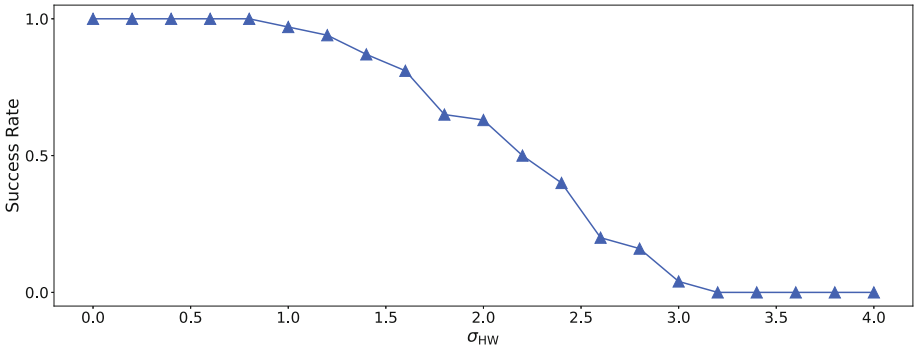



**Fig. 5.** Success rate on an optimized implementation

The number of traces is also a significant determinant in the success rate of DC attacks. Hence, we conduct experiments on optimized implementation and show the outcomes of variable number of traces leading to successful attacks in Fig. 6. We see that passing more traces beneath each noise can result in a higher success rate. Note, however, that our earlier results indicate that with higher noise (i.e. about $\sigma_{\mathrm{HW}} > 2.5$) is not possible to achieve a better success rate with a large number of traces.

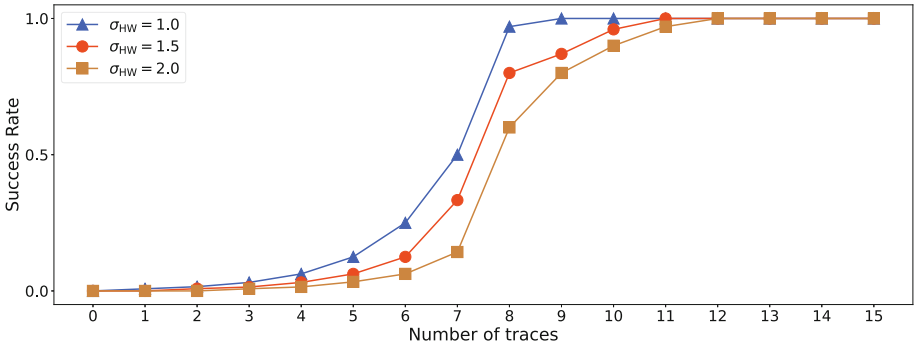



**Fig. 6.** Relationship between number of traces and success rate under different noises

## 5 Conclusion and Open Problems

In this work, we perform an efficient SASCA on Ascon-128 by constructing the optimized factor graph. Then the simulations are run on an 8-bit platform. We can efficiently exploit the leakage information of all the leaking operations in implementation (compared to DC attacks). Our attacks are ideal in terms of data complexity, time, and memory efficiency. At the same time, there is a lot of room for improvement in our work. For model selection, we use a simple Hamming-weight leakage model. More complicated profiling tools, such as multivariate value templates or machine-learning algorithms, are expected to improve performance, allowing attacks in more scenarios. Besides, in future work, we look forward to demonstrating the feasibility of experiments on more platforms (such as 32-bit). Recovering the entire key using only the information from a single trace is another interesting scope for additional research.

The attack requires only a small number of traces and can be successful in high noise levels. As a result, similar attacks can be applied to IoT devices using Ascon encryption, which might recover the entire key after several runs. We therefore consider that unprotected implementations cryptographic algorithms should be avoided whenever possible, even when long-term encryption is not involved.

**Acknowledgements.** This work was supported by the National Key R&D Program of China (Grant No. 2020AAA0107703), the National Natural Science Foundation of China (Grant No. 62132008, 62072247, 62071222), the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20220075).

## References

1. Adomnicai, A., Fournier, J.J., Masson, L.: Masking the lightweight authenticated ciphers acorn and ascon in software. Cryptology ePrint Archive (2018)
2. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Keccak specifications. Submission to NIST (round 3) (2011). https://keccak.team
3. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1. 2. Submission to the CAESAR competition (2016)
4. Gallager, R.: Low-density parity-check codes. IRE Trans. Inf. Theory **8**(1), 21–28 (1962)
5. Gross, H., Wenger, E., Dobraunig, C., Ehrenhöfer, C.: Ascon hardware implementations and side-channel evaluation. Microprocess. Microsyst. **52**, 470–479 (2017)
6. Kannwischer, M.J., Pessl, P., Primas, R.: Single-trace attacks on keccak. IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 243–268 (2020)
7. MacKay, D.J., et al.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)
8. Martin, S., Ferdinand, B.: Reference, optimized, masked C and ASM implementations of Ascon. https://github.com/ascon/ascon-c

9. Nikova, S., Rechberger, C., Rijmen, V.: Threshold implementations against side-channel attacks and glitches. In: International Conference on Information and Communications Security, pp. 529–545. Springer, Berlin, Heidelberg (2006). https://doi.org/10.1007/11935308_38
10. Pearl, J.: Reverend Bayes on inference engines: a distributed hierarchical approach. In: Probabilistic and Causal Inference: The Works of Judea Pearl, pp. 129–138 (2022)
11. Pessl, P., Primas, R.: More practical single-trace attacks on the number theoretic transform. In: Schwabe, P., Thériault, N. (eds.) LATINCRYPT 2019. LNCS, vol. 11774, pp. 130–149. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30530-7_7
12. Renauld, M., Standaert, F.-X.: Algebraic side-channel attacks. In: Bao, F., Yung, M., Lin, D., Jing, J. (eds.) Inscrypt 2009. LNCS, vol. 6151, pp. 393–410. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16342-5_29
13. Samwel, N.: Side-channel analysis of keccak and ascon (2016)
14. Samwel, N., Daemen, J.: DPA on hardware implementations of ascon and keyak. In: Proceedings of the Computing Frontiers Conference, pp. 415–424 (2017)
15. Veyrat-Charvillon, N., Gérard, B., Standaert, F.-X.: Soft analytical side-channel attacks. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8873, pp. 282–296. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45611-8_15

# Privacy-preserving WiFi Fingerprint Localization Based on Spatial Linear Correlation

Xu Yang, Yuchuan Luo(✉), Ming Xu, Shaojing fu, and Yingwen Chen

National University of Defense Technology, Changsha 410008, China
luoyuchuan09@nudt.edu.cn

**Abstract.** With the widespread deployment of IoT (Internet of Things) devices, WiFi fingerprint-based localization is becoming one of the most promising techniques for indoor localization. A client is able to obtain its location by providing its measured fingerprint (vector of WiFi signal strengths) to the service provider who maps the fingerprint against the database and returns the result back to the client. However, traditional applications of WiFi fingerprint-based localization may disclose the client's location privacy and often incur high consumption of communication and computing resources. In this paper, we focus on implementing a privacy-preserving framework with high efficiency and accuracy for WiFi fingerprint-based localization. Firstly, to reduce computational overhead at the server side, we introduce a clustering algorithm called k-means++ in offline phase. Besides, we explore the correlation of the fingerprint and propose a Pearson correlation based distance computation method, which achieves better accuracy than traditional Euclidean distance. Finally, we secure the overall computation by adapting a series of secure multi-party computing primitives. Theoretical analysis is carried out to prove the security of our scheme. Experiments on real-world datasets indicate that our scheme achieves better practicality and efficiency compared with existing methods. Compared to existing work PriWFL and PPWFL, our scheme reduces the average distance error by approximately 4.5% and 2.9% under a query time of less than 0.2$s$.

**Keywords:** WiFi fingerprint-based localization · Privacy-preserving · Secret sharing · Pearson correlation

## 1 Introduction

A Location-Based Service (LBS) is a service for clients to obtain additional information by providing their physical locations. However, in places like malls and tunnels where GPS signals are not available, such service is hard to deploy. To solve this problem, WiFi fingerprint-based localization is proposed [1]. It

This work was supported in part by the National Nature Science Foundation of China (No. 62102429, 62102422, 62072466, 61872372), and the NUDT Grants (No. ZK19-38).


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 401–412, 2022.
https://doi.org/10.1007/978-3-031-19208-1_33

leverages existing WiFi Access Points (APs) which can be found everywhere easily. From those APs, the client reads Received Signal Strengths (RSSs) and sends them to a server that holds a database of collected RSSs and associated locations (so called Reference Points, RPs). By mapping the client's RSSs against all RPs in the database using similarity calculation algorithm, $k$ best matched locations are obtained and returned to the client.

**Challenges.** However, applying the above method in real-world applications has not unleashed its full potential yet due to some limitations as follows.

(1) *The accuracy of fingerprint-based localization is not stable enough.*

Due to the complex propagation phenomena (e.g. multipath fading, shadowing) of radio signals in indoor environment, most existing works cannot achieve acceptable performance caused by fluctuations in measured RSSs values with different devices and different times conditions. Sun *et al.* [2] improved the accuracy of indoor localization in a specific environment, but the localization stability is poor for different terminals due to different hardware and wireless technologies used. Chang *et al.* [3] proposed to use the Signal Strength Difference (SSD) of different APs at the same location to alleviate environmental interference. It can reduce the impact of terminal heterogeneity to a certain extent but also lead to a decrease in accuracy.

(2) *There is a risk of privacy leakage in existing works.*

In traditional WiFi fingerprint-based localization framework, the client uploads the plaintext of its measured RSSs vector to the server who then maps the data against the database. The first problem is that the query vector contains the client's location information. Existing research indicates that malicious servers could even infer a client's social relationships from its query history [4]. Another problem is that malicious clients may infer information from the database at the server side. Existing research [5] indicates that malicious clients could restore the whole database through well-designed query. Yang *et al.* [6] explored a method to protect location privacy by k-annoymity. Unfortunately, these methods require a trusted third party so that they are not widely applied.

**Our Contributions.** In this paper, we propose a novel scheme for WiFi fingerprint-based localization which can not only solve these two problems, but also improve the efficiency to make it practical to the real-world applications.

- We investigate the correlation of the fingerprint and propose a Pearson correlation based distance computation method, which achieves better accuracy than traditional Euclidean distance. To better accelerate query execution, we also introduce secure k-means++ algorithm to preprocess the database at the server side, which can help divide fingerprint database into several clusters.
- To protect the client's data privacy, we design an efficient outsourcing protocol using two semi-trusted third parties as servers. Additive secret sharing is applied to encrypt the fingerprints, which can prevent the sensitive information from being stolen. To complete offline and online computations, we modify the original algorithms and obtain secure k-means++ and secure Pearson correlation algorithms.

• We implement a novel scheme for WiFi fingerprint-based localization. Theoretical analysis and experiments on real-world datasets show that our method can achieve higher efficiency and accuracy than existing schemes while still protecting privacy.

The rest of the paper is organized as follows. Section 2 introduces the system model and the prelinminaries. Section 3 details our proposed scheme. We discuss the performance in Sect. 4 and conclude this paper in Sect. 5.

## 2 Problem Formulation

### 2.1 System and Threat Model

In this paper, we focus on implementing a privacy-preserving framework for WiFi fingerprint-based localization. As shown in Fig. 1, there are five entities in our scheme: the data collector $\mathcal{S}_0$, two servers $\mathcal{S}_1$ and $\mathcal{S}_2$, the client $\mathcal{U}$, and the key generator $\mathcal{KG}$. The data collector $\mathcal{S}_0$ collects data and outsources them to $\mathcal{S}_1$ and $\mathcal{S}_2$ after encrypting (see 2.2). Then $\mathcal{S}_1$ and $\mathcal{S}_2$ start to perform offline calculations to preprocess the data. Once $\mathcal{U}$ submits a new query request, they collaboratively perform a series of computation. $\mathcal{KG}$ is introduced to generate random values which would be sent to $\mathcal{S}_1$ and $\mathcal{S}_2$. $\mathcal{U}$ would receive and recover the encrypted result which can be applied to various localization applications.

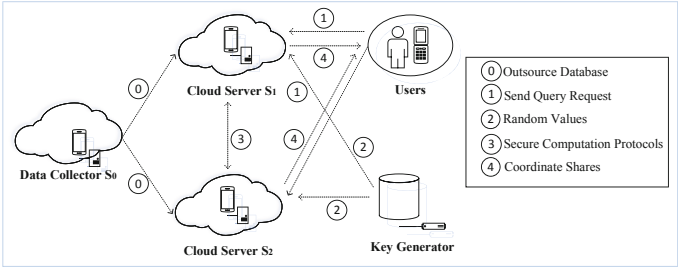



**Fig. 1.** System Model of Proposed Scheme

In this article, $\mathcal{S}_1$ and $\mathcal{S}_2$ are both honest but curious which means that they are curious to infer sensitive information but will not collude with each other. The key generator is considered as a lightweight server which has few computing resources, and it is also honest-but-curious. Our goal is to implement privacy-preserving fingerprint-based localization with high accuracy and efficiency.

### 2.2 Prelinminaries

**Traditional WiFi Fingerprint-based Localization.** Traditional WiFi fingerprint localization scheme has two phases. In offline phase, in order to construct the fingerprint database $< i, V_i, (x_i, y_i) >$, the data collector needs to

collect all the AP signals for each location $i$, where $(x_i, y_i)$ is the physical coordinate for location $i$ and $V_i$ is the corresponding RSSs vector. Each AP signal corresponds to one entry in the fingerprint, so $V_i = (v_1, v_2, ..., v_k, ..., v_M)$ is an $M$-dimensional vector. The data collector stores the database locally and ensures the security of the database. In online phase, the client constructs a fingerprint vector $F = (f_1, f_2, ..., f_k, ..., f_M)$ by measuring AP signals, and then it was encapsulated into a query request and send to the service provider. According to Eq. (1), the service provider calculates the similarities between the client's query vector and RSSs in the database. The service provider selects $K$ results with the smallest distance values and returns the coordinates to the client.

$$d_i = ||F - V_i||^2 = \sum_{j=1}^{M} (v_{i,j} - f_j)^2 \tag{1}$$

**Additive Secret Sharing.** In multi-party secure computing [7], secret sharing is one of the most important technologies and has various application scenarios. Unlike homomorphic encryption [8], secret sharing randomly splits the original data $x$ into multiple shares $x_1, x_2, ..., x_n$ over a finite ring $Z_q, q \in N^*$, and we use $x_i$ or $[x]_i$ to represent the shared value of $x$. After collecting and summing all shares over the finite ring, the original data can be recovered. In our study, $n$ is set as 2 which means that the original data is split into $x_1$ and $x_2$, which are stored on two servers respectively. In the whole calculation process, each server has only one share of the data, and cannot infer any valuable information.

In our study, addition and subtraction of scalars can be done locally on servers. Specifically, two servers hold shared values $x_1, y_1$ and $x_2, y_2$ respectively, and they only need to calculate $x_i \pm y_i$ where $i = 1, 2$. As for scalar multiplications $SecMul$, Beaver's Triplet [9] is introduced to pre-generate random triples $a_i$, $b_i$ and $c_i$ where $i = 1, 2$ and $c_1 + c_2 = (a_1 + a_2) \times (b_1 + b_2)$. Through several steps of local computation and communication, they get $z_i$ where $z_1 + z_2 = x \times y$. Furthermore, we directly borrow ideas of $SecMatMul$, $SecMatInv$, $SecDiv$, $SecCmp$ and $SecSort$ from Xia *et al.*'s work [10], which can help servers obtain the product of two matrixs, the inverse of a matrix, the divsion of two numbers, the sorting of two numbers and the sorting of a series of numbers respectively.

## 3 Privacy-preserving WiFi Localization Framework

We implement privacy-preserving WiFi fingerprint localization in two phases: the offline phase and the online phase. Firstly, the servers collect RSSs and locations to form a database locally. When a client submits a query request, the servers begin to perform online computations. Our goal is to allow clients to quickly obtain their location without leaking their sensitive information.

### 3.1 Offline Phase

Zheng *et al.* [11] found that RSSs have a significant spatial linear correlation, which is the focus point in our study. Assuming that there are $d$ APs in the target

environment, the RSSs of a mobile device collected for the $n$th time at the location $l$ is represented as a vector $S_{l,n} = \{rss_{l,n,1}, rss_{l,n,2}, ..., rss_{1,n,p}, ..., rss_{l,n,d}\}$, where $rss_{l,n,p}$ represents the signal strength sent by the $p$th AP. After some filtering of the RSS collected at the reference point $l$, the RSS sample matrix $F_l$ are obtained, which are represented as Eq. (2).

$$F_l = \begin{bmatrix} rss_{l,1,1} & rss_{l,2,1} & \dots & rss_{l,N,1} \\ rss_{l,1,2} & rss_{l,2,2} & \dots & rss_{l,N,2} \\ \dots & \dots & rss_{l,n,p} & \dots \\ rss_{l,1,d} & rss_{l,2,d} & \dots & rss_{l,N,d} \end{bmatrix}_{d \times N} \tag{2}$$

To fully utilize the spatial correlation between RSSs, the fingerprint for a specific location should be collected multiple times. Therefore, the client should also collect multiple sets of fingerprints in real time, so that the similarity calculation can be performed correctly. However, collecting multiple sets of fingerprint data online can be very slow for a resource-constrained device. To achieve real-time localization, the matrix $F_l$ needs to be aggregated. Specifically, we split $F_l$ into $M$ sub-matrices equally by column. The value of $M$ needs to be the same as the number of the RSSs vector collected by the client during the online phase. In our experiment, $M$ is set as 3 to prevent the collection of fingerprints from taking up too much time. Then, we average each row of the submatrix which converts the submatrix into a vector with dimension $d \times 1$. Finally, we obtain $F_l^{'} = [\overline{RSS}_{l,1}, \overline{RSS}_{l,2}, ..., \overline{RSS}_{l,m}, ..., \overline{RSS}_{l,M}]_{d \times M}$.

As shown in Fig. 2, $\mathcal{S}_0$ would outsource the database $F'$ to server $\mathcal{S}_1$ and $\mathcal{S}_2$ using additive secret sharing. After servers $\mathcal{S}_1$ and $\mathcal{S}_2$ obtain the shared value of $F_l^{'}$, due to its high dimensionality and large size, it is difficult to perform similarity calculation directly. In this study, a privacy-preserving method for data dimension reduction and data clustering is introduced. By using modules in 2.2, we apply secure dimensionality reduction algorithm called secure PCA [10] and each server obtains $W_i$ as its result after dimensionality reduction.

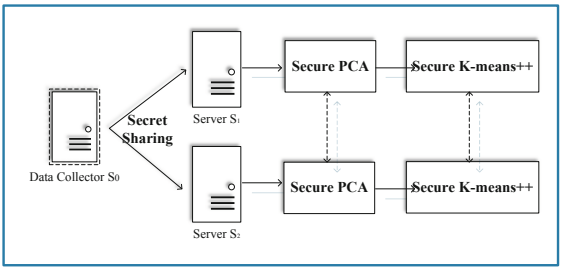



**Fig. 2.** Offline Workflow

Then, we explore the distribution of the fingerprint database and propose a secure k-means++ based clustering method, which achieves better convergence speed and clustering accuracy than the normal k-means algorithm [12].

By applying secure k-means++, each fingerprint is allocated to a cluster with a center $\boldsymbol{CL}_i$, as shown in algorithm 1. From line 2 to line 15, the algorithm selects initial centers randomly from all fingerprints. From line 16 to line 22, the algorithm iteratively finds better centers, which reduces the clustering error. These results are kept in the servers as shared value so the privacy information would not be leaked. It should be noted that the number of clusters $K$ is an important hyperparameter. If it is too large, there would be too many clusters and so the error would increase. Besides, the server would roughly estimate the client's location through the cluster, which is against our privacy goal. If $K$ is too small, there are still many fingerprints in each cluster, which would not help accelerate online computation. In the following experiment, we set $5 \leq K \leq 10$.

**Algorithm 1.** Secure k-means++ protocol

**Input:** 1: $S_i$ has k, one share of vectors $\{\boldsymbol{W}_i^l\}$ and the $ID$ each vector corresponds, here $l \in [1, n]$, n is the number of vectors.

**Onput:** 2: $S_i$ gets one share of cluster centers, and the information the vectors in each cluster.

1: $\mathcal{KG}$ generates enough random numbers and matrices the sub-protocol uses and sends to $S_i$.
2: $\mathcal{S}_1$ randomly picks a vector as the first cluster center $\{CL_1\}$ and sends the corresponding $ID$ to $\mathcal{S}_2$.
3: **for** $l = 1 : n$ **do**
4: $\quad\mathcal{S}_1\&\mathcal{S}_2$ collaboratively compute the squared Euclidean distances between $\boldsymbol{W}_i^l$ and all the center vectors based on $SecMatMul$.
5: $\quad\mathcal{S}_1\&\mathcal{S}_2$ collaboratively seek the nearest center for $\boldsymbol{W}_i^l$ and obtain the distance $d_i^l$ based on $SecSort$.
6: **end for**
7: $\mathcal{S}_1\&\mathcal{S}_2$ collaboratively compute the sum of all distance $\sum_{l=1:n} d_i^l$ and obtain $p^l = \frac{d^l}{\sum_{l=1:n} d^l}$ using $SecDiv$ where $p$ is also a shared value. $\mathcal{S}_1$ randomly picks a number $c \in [0, 1]$, and sends it to $\mathcal{S}_2$.
8: **for** $l = 1 : n$ **do**
9: $\quad\mathcal{S}_1\&\mathcal{S}_2$ collaboratively compare $\sum_{k=1:l} p^k$ with $c$ using $SecSort$.
10: $\quad$**if** $c$ is smaller **then**
11: $\quad\quad\mathcal{S}_1\&\mathcal{S}_2$ picks the corresponding vector $W^l$ as the next cluster center.
12: $\quad\quad$break
13: $\quad$**end if**
14: **end for**
15: repeat line 3 to line 14 until there are $K$ centers.
16: **for** $l = 1 : n$ **do**
17: $\quad\mathcal{S}_1\&\mathcal{S}_2$ collaboratively compute the squared Euclidean distances between $\boldsymbol{W}_i^l$ and centroid vectors based on $SecMatMul$.
18: $\quad\mathcal{S}_1\&\mathcal{S}_2$ collaboratively seek the nearest centroid vector of $\boldsymbol{W}_i^l$ based on $SecSort$.
19: $\quad S_i$ puts the $\boldsymbol{W}_i^l$ in the category which nearest centroid represents.
20: **end for**

21: $S_i$ computes the mean value of vectors in each category, which is actually the share of new centroid vectors $\boldsymbol{CL_i}$.
22: Repeat line 16 to line 21 until a certain rounds or $\{\boldsymbol{CL_i}\}$ unchanging.

### 3.2 Online Phase

If a client tries to obtain its locations, it would send $R_i$ to servers $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively, where $R$ is an RSSs vector receiving in real time. Since the fingerprint database has been preprocessed by the secure PCA algorithm and the secure k-means++ algorithm, so the new query vector also needs to be dimensionally reduced and to find its cluster, as shown in Fig. 3. Specifically, the client firstly collects RSSs for $M$ times and put them into a matrix as shown in Eq. (3).

$$R = \begin{bmatrix} rss^*_{1,1} & rss^*_{2,1} & \dots & rss^*_{M,1} \\ rss^*_{1,2} & ss^*_{2,2} & \dots & rss^*_{M,2} \\ \dots & \dots & rss^*_{m,p} & \dots \\ rss^*_{1,d} & rss^*_{2,d} & \dots & rss^*_{M,d} \end{bmatrix}_{d \times M} . \tag{3}$$

By performing the same computation as in 3.1, we could obtain the dimension-reduced query vector $R'$. After that, we calculate the distance between $R'$ and the cluster centers obtained by secure k-means algorithm, and then we obtain a cluster of fingerprints with a center that has smallest distance.

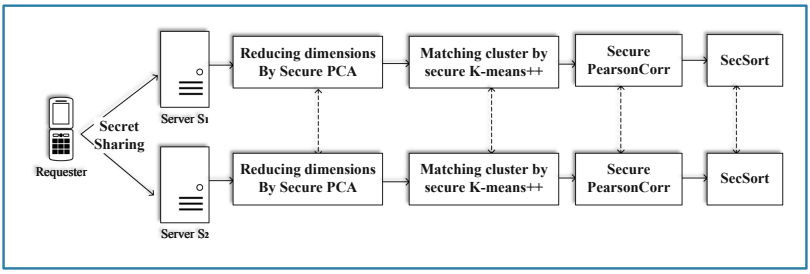



**Fig. 3.** Online Workflow

$$PearsonCorr = \frac{\sum_{n=1}^{M}\sum_{p=1}^{d}(rss_{l,n,p} - \mu^l)(rss^*_{n,p} - t)}{\sqrt{\sum_{n=1}^{M}\sum_{p=1}^{d}(rss_{l,n,p} - \mu^l)^2}\sqrt{\sum_{n=1}^{M}\sum_{p=1}^{d}(rss^*_{n,p} - t)^2}} \tag{4}$$

Next, we calculate the similarity between the query vector $R'$ and each fingerprint $W'$ in the database. Here we apply Pearson correlation coefficient [13] instead of Euclidean distance because it can make better use of the spatial correlation between RSSs, as shown in Eq. (4). By adapting secret sharing, we secure

**Algorithm 2.** Secure PearsonCorr

**Input:** $S_i$ has $W_i^l$ and $R_i'$.
**Onput:** $S_i$ gets the Pearson correlation between inputs.

1: $\mathcal{KG}$ generates enough random numbers the sub-protocol uses and sends to $S_i$.
2: $S_i$ compute $[u^l]_i = \frac{\sum_{n=1}^{M}\sum_{p=1}^{d}[rss_{l,n,p}]_i}{d\times M}$, $[t]_i = \frac{\sum_{n=1}^{M}\sum_{p=1}^{d}[rss^*_{n,p}]_i}{d\times M}$,
$[rss_{l,n,p}]_i \leftarrow [rss_{l,n,p}]_i - [u^l]_i$, $[rss^*_{n,p}]_i \leftarrow [rss^*_{n,p}]_i - [t]_i$.
3: $\mathcal{S}_1\&\mathcal{S}_2$ collaboratively compute $[r_1]_i = \sum_{i=1}^{M}\sum_{j=1}^{d} SecMul([rss_{l,n,p}]_i, [rss^*_{n,p}]_i)$,
$[r_2]_i = \sum_{i=1}^{M}\sum_{j=1}^{d} SecMul([rss_{l,n,p}]_i, [rss_{l,n,p}]_i)$ and
$[r_3]_i = \sum_{i=1}^{M}\sum_{j=1}^{d} SecMul([rss^*_{n,p}]_i, [rss^*_{n,p}]_i)$.
4: $\mathcal{S}_1\&\mathcal{S}_2$ collaboratively compute $[r_1]_i \leftarrow SecMul([r_1]_i, [r_1]_i)$ and
$[r_4]_i = SecMul([r_2]_i, [r_3]_i)$.
5: $\mathcal{S}_1\&\mathcal{S}_2$ collaboratively compute $[res]_i = SecDiv([r_1]_i, [r_4]_i)$.
6: return shared values $[res]_i$.

the similarity computation without revealing the privacy of client's location, as shown in algorithm 2. After obtaining all the similarities between the query vector and fingerprint in the database, $\mathcal{S}_1$ and $\mathcal{S}_2$ collaboratively sort all these results by using *SecSort* and then return the best-matched coordinates with the minimum similarity value to the client.

## 4 Evaluation

### 4.1 Security Analysis

By using universal composability framework, traditional secret sharing is proved to be secure. If there is a simulator $\mathcal{S}$ that can simulate indistinguishable view to that in the real world for clients, then protocol is regarded as secure. The following definitions are needed for the complete analysis:

**Definition 1.** If a simulator $\mathcal{S}$ can generate a computationally indistinguishable view for the adversary in the real world within probabilistic polynomial-time, then the protocol is regarded as secure.

**Definition 2.** A protocol is theoretically simulatable if all of its sub-protocol are simulatable.

**Definition 3.** In honest-but-curious models, protocols including $SecMatMul$, $SecMatInv$, $SecDiv$, $SecCmp$ $SecSort$ and $SecurePCA$ are secure.

For more detailed proofs, readers can refer [10,14,15] for these definitions. With the above definitions, we can conclude that secure k-means and secure PearsonCorr are both simulatable since they are all constructed by subprotocols like $SecMatMul$, $SecMatInv$ $SecDiv$ and so on.

### 4.2 Performance Evaluation

To evaluate the performance of our scheme, we conduct experiments on real-world datasets and compare with two other works called PriWFL [8] and PPWFL [16]. Both of them protect privacy while implementing the localization task. Specifically, PriWFL uses Semi-Homomorphic Encryption (SHE) while the PriWFL uses Secure Multiparty Computation (SMC) instead. Both of them performs computation on a central server which also stores the fingerprint database. Compared with these two works, our method shows better accuarcy and efficiency.

We build two servers in our experiment with each of them equipped with a 32-core Intel Xeon CPU @ 3.40GHz and 64GB RAM. The algorithms are programmed with Python 3.6 and the programs run on Windows 10. Simulation is run on three types of datasets including UJIIndoorLoc Dataset (UIJ), BLE RSSI Dataset (BLE) and Wireless Indoor Localization Dataset (WLI). UIJ contains more than 20,000 samples with 529 feature dimensions, while BLE contains 6661 samples with 15 feature dimensions, and WLI contains 2000 samples with and 7 feature dimensions. It should be noted that most data in UIJ dataset are null, so the dimension with smaller variance is removed and only 92 feature dimensions remains. Besides, we only use 2000 records of UIJ measured at the same building and the same floor. Also, We regard null values in all records as 0, and invert all negative numbers. The dataset contains training and test sets, and each record contains an RSSs vector and corresponding physical coordinates.

• **Efficiency Evaluation**

In this section, we apply real-world datasets to each of the three schemes, then record the average query time. We randomly sample from datasets to build the fingerprint databases that need to be stored on the server side and the size varies from 100 to 700. As the size of the database becomes larger, the accuracy would become higher. However, this would also generate a large computing overheads at the server side and slow down the query speed. There are some hyperparameters in our protocol that have a strong impact on the execution of the algorithm. For example, in the secure k-means++ algorithm, $K$ is the number of the clusters. Increasing $K$ would lead to increased accuracy and decreased level of privacy. In the secure PCA algorithm, $s$ is the number of eigenvectors taken after eigenvalue decomposition of the covariance matrix, and larger $s$ would result in fewer feature dimensions being filtered out. Also, the parameter $M$ determines how many times we collect fingerprints for one location and how many times the client measures RSSs in the online phase. In the subsection, we fix $K = 5$, $s = 30$, and $M = 3$.

As shown in Fig. 4, our scheme has advantages in query efficiency compared to PriWFL and PPWFL. Taking the WLI result as an example, when 700 random samples are used as the fingerprint database at the server side, PriWFL takes an average of 0.971s to query, while PPWFL takes 0.198s, and our method only takes 0.181s. In the experiments using the UIJ dataset, when the number of fingerprints in the database increases from 100 to 700, the average query time of PriWFL increases from around 1s to around 3.5s, but PPWFL and our method only takes around 0.1s. Experiments on BIE and UIJ have similar results.

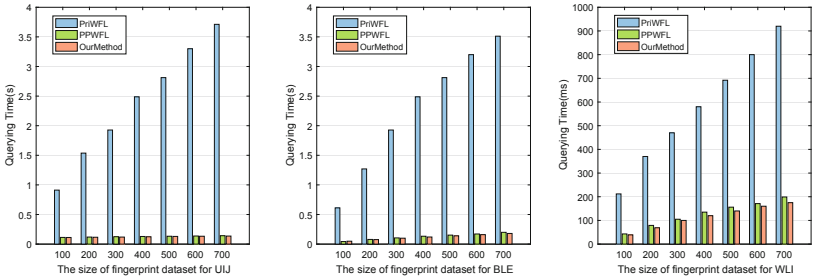



**Fig. 4.** Average quering time under different size of fingerprint dataset

- **Accuracy Evaluation**

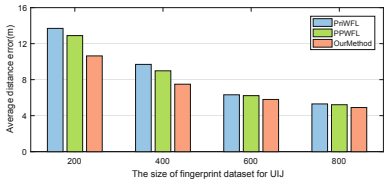



**Fig. 5.** Average distance error

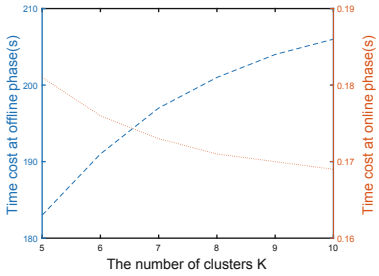



**Fig. 6.** Time cost as K changes

Due to the time-varying feature of RSSs and the difference for different terminals in signal acquisition, improving accuracy is the main challenge currently. Based on the linear correlation of RSS in space, we propose the secure Pearson algorithm. The experimental results are obtained in the following, and we set other parameters the same as in efficiency evaluation. Table 1 shows the average distance errors of these three methods on different datasets when the sample size reaches 800. It is easy to see that the error of PriWFL is always the largest and that of our method is the smallest. When UIJ dataset is applied, the average distance error of our method is reduced by approximately 4.5% and 2.9%. Besides, the average error is measured with different fingerprint database size. As shown in Fig. 5, when the dataset is UIJ and the sample size is 200, the average error of our method is 13.3m, which is much smaller than 17.12m of PriWFL and 16.13m of PPWFL. As the size of the database increases, the error of our method remains the smallest, but the advantage reduces slightly. Hence, we can conclude that our method can show more advantages in accuracy when the size of database is not too large.

- **Clustering**

In the offline phase, secure k-means++ algorithm splits the fingerprint database into different clusters before querying. When a query request is sub-

**Table 1.** Average Distance Error When Sampling Size is 800

| | PriWFL | PPWFL | OurMethod |
|---|---|---|---|
| UIJ | 6.63m | 6.52m | 6.33m |
| BLE | 8.02m | 7.92m | 7.84m |
| WLI | 6.40m | 6.33m | 6.27m |

mitted to the server, the server firstly calculates the distance between the query vector and the center of all clusters, and selects the cluster with the smallest distance value which might contains the best-matched fingerprint. The number of clusters is an important hyperparameter, because if it is too large, then the cluster obtained by executing secure k-means++ algorithm may disclose privacy information related to the client's location. If it is too small, it would hardy help the later online computation. Therefore, we need to make a trade-off between privacy and efficiency.

As shown in Fig. 6, $K$ varies in $[5, 10]$, and record the online time and offline time respectively (the offline time only means the processing time of executing secure k-means++ algorithm). Other settings remain the same as in efficiency evaluation, and the size of the fingerprint database is set to be 600. We can see that as the number of clusters increases, the offline execution time increases, and the online time decreases significantly. This result shows that the secure k-means++ algorithm help reduce the computational overheads for the servers.

## 5 Conclusion

In this paper, we focus on implementing privacy preserving WiFi fingerprint-based localization for the resource-constrained IoT devices. To prevent the leakage of location privacy, we use an efficient outsourcing protocol using two distributed servers. To reduce computation overheads at the server side in online phase, we design novel algorithms called secure k-means++ to help cluster the data. Besides, we introduce Pearson correlation to replace Euclidean distance, which makes the query result more stable and accurate. We thoroughly analyze the security of our protocols and conduct experiments over real-world datasets to record the efficiency and accuracy. The result indicates that our method have better performance than existing schemes while still protecting location privacy.

## References

1. Lui, G., Gallagher, T., Li, B., Dempster, A.G., Rizos, C.: Differences in RSSI readings made by different Wi-Fi chipsets: a limitation of WLAN localization. In: 2011 International Conference on Localization and GNSS (ICL-GNSS), pp. 53–57 (2011). https://doi.org/10.1109/ICL-GNSS.2011.5955283
2. Sun, Y.L., Xu, Y.B.: Error estimation method for matrix correlation-based Wi-Fi indoor localization. KSII Trans. Internet Inf. Syst. **7**(11), 2657–2675 (2013). https://doi.org/10.3837/tiis.2013.11.006

3. Chang, N., Rashidzadeh, R., Ahmadi, M.: Robust indoor positioning using differential Wi-Fi access points. IEEE Trans. Consum. Electron. **56**(3), 1860–1867 (2010). https://doi.org/10.1109/TCE.2010.5606338
4. Shokri, R., Theodorakopoulos, G., Le Boudec, J.Y., Hubaux, J.P.: Quantifying location privacy. In: 2011 IEEE Symposium on Security and Privacy, pp. 247–262 (2011). https://doi.org/10.1109/SP.2011.18
5. Yang, Z., Järvinen, K.: The death and rebirth of privacy-preserving WiFi fingerprint localization with Paillier encryption. In: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications. pp. 1223–1231 (2018). https://doi.org/10.1109/INFOCOM.2018.8486221
6. Yang, D., Fang, X., Xue, G.: Truthful incentive mechanisms for k-anonymity location privacy. In: 2013 Proceedings IEEE INFOCOM, pp. 2994–3002 (2013). https://doi.org/10.1109/INFCOM.2013.6567111
7. Zheng, X., Cai, Z.: Privacy-preserved data sharing towards multiple parties in industrial IoTs. IEEE J. Sel. Areas Commun. **38**(5), 968–979 (2020)
8. Li, H., Sun, L., Zhu, H., Lu, X., Cheng, X.: Achieving privacy preservation in WiFi fingerprint-based localization. In: IEEE INFOCOM 2014 - IEEE Conference on Computer Communications. pp. 2337–2345 (2014). https://doi.org/10.1109/INFOCOM.2014.6848178
9. Beaver, D.: Efficient multiparty protocols using circuit randomization. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 420–432. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_34
10. Xia, Z., Gu, Q., Xiong, L., Zhou, W., Weng, J.: Privacy-preserving image retrieval based on additive secret sharing. ArXiv abs/2009.06893 (2020)
11. Zheng, Z., Chen, Y., He, T., Li, F., Chen, D.: Weight-RSS: a calibration-free and robust method for WLAN-based indoor positioning. Int. J. Distrib. Sens. Netw. **11**(4), 573582 (2015)
12. Arthur, D., Vassilvitskii, S.: K-means++: the advantages of careful seeding. In: SODA '07 (2007)
13. Biber, D.: Pearson correlation coefficients for all linguistic features, p. 270–279. Cambridge University Press (1988). https://doi.org/10.1017/CBO9780511621024.013
14. Bogdanov, D., Laur, S., Willemson, J.: Sharemind: a framework for fast privacy-preserving computations. In: Jajodia, S., Lopez, J. (eds.) ESORICS 2008. LNCS, vol. 5283, pp. 192–206. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88313-5_13
15. Huang, K., Liu, X., Fu, S., Guo, D., Xu, M.: A lightweight privacy-preserving CNN feature extraction framework for mobile sensing. IEEE Trans. Dependable Secure Comput. **18**(3), 1441–1455 (2021). https://doi.org/10.1109/TDSC.2019.2913362
16. Wu, W., Fu, S., Luo, Y.: Practical privacy protection scheme in WiFi fingerprint-based localization. In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), pp. 699–708 (2020). https://doi.org/10.1109/DSAA49011.2020.00080

# Secure RFID Handwriting Recognition–Attacker Can Hear but Cannot Understand

Qihang Zhang, Jiuwu Zhang, Xiulong Liu(✉), Xinyu Tong, and Keqiu Li

College of Intelligence and Computing, Tianjin University, Tianjin, China
xiulong_liu@tju.edu.cn

**Abstract.** Radio Frequency Identification (RFID) has been adopted in various applications owning to its many attractive properties such as low cost, no requirement on line-of-sight, and battery-free. This paper studies the problem of RFID-based Handwriting recognition, which is practically important in Human-Computer Interaction (HCI) scenarios. To the best of our knowledge, the state-of-the-art works beget leaking user privacy, because the malicious attacker can eavesdrop on the RFID signals (*e.g.*, tag phase) broadcast in the air and further analyze the user's handwriting activity. To address the privacy leakage issue, we propose a secure RFID handwriting recognition system named SecRFPen to enable privacy-preserving handwriting recognition. In SecRFPen, the legal reader switches the probing frequency and power, the phase angles of RF signals reflected by the tagged pen will change accordingly. Thus, the phase profile of the tagged pen is actually determined by both reader-tag hardware characteristics and handwriting movements. We propose an authentication matrix to quantify RFID device hardware characteristics, which can be measured by legal users in advance. Thus, the legal RFID reader can recognize the handwriting activity via analytics on the authentication matrix and tag phase profile. On the contrary, since the malicious attacker knows nothing about the hardware characteristics of legal RFID devices, it cannot understand handwriting even if it can hear the tag signals. We implement the SecRFPen system based on the Commercial-Off-The-Shelf (COTS) RFID devices. Extensive experimental results demonstrate that the recognition accuracy of legal users can reach 94.2%, while the recognition accuracy of the malicious attacker is as low as 35.1%.

**Keywords:** RFID · Handwriting recognition · Privacy protection

## 1 Introduction

Handwriting recognition is an important application of wireless sensing technology, which can be deployed in many intelligent scenarios such as smart homes and smart supermarkets. Generally, the state-of-the-art works usually employ cameras [1], acoustic signals [2], intelligent sensors [3] and wireless signals [4] to


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 413–426, 2022.
https://doi.org/10.1007/978-3-031-19208-1_34

realize handwriting recognition. Among these techniques, RFID-based handwriting recognition attracts great attention, as RFID has many attractive properties including simultaneous identification of multiple targets, no requirement on line-of-sight, battery-free, and low cost of tags, *etc.*. In recent years, a series of RFID handwriting recognition systems (*e.g.,* RF-IDraw [5], RF-copybook [6], RF-Pen [4]) were proposed. Despite their high recognition accuracy, these systems do not take *user privacy* into consideration, which leaves the malicious attacker a chance to eavesdrop on the user's handwriting activities. Specifically, the above existing systems commonly leverage the phase data to recognize the handwriting movements, because the received tag phase is highly related to the moving trajectory of tags. However, besides the legal reader, the malicious attacker can also eavesdrop the RFID signals (e.g., tag phase) broadcast in the air and further analyze the user's handwriting activity. Hence, the existing works have the risks of leakage of user privacy. To this end, this paper aims at designing a secure RFID handwriting recognition system named SecRFPen.

The basic idea of SecRFPen is to analyze the phase data, so that we can recognize fine-grained handwriting activities. A key point we need to consider is how SecRFPen can prevent handwriting information leakage by encrypting phase data. Specifically, we propose a method to encrypt the phase data by hopping frequency and power. When the system works, the data collection program will switch random power and frequency in short intervals, which will cause the collected phase data to seem messy, thus the malicious attacker cannot extract useful information from the collected data. After data collecting, legal users can recover the phase data by the proposed algorithm and use the recovered phase data to train the recognition model, resulting in high-accuracy handwriting recognition. In contrast, the malicious attacker cannot correctly recover the phase data and thus cannot recognize the handwriting activity.

To build SecRFPen system, the following two challenges need to be addressed.

*The first challenge is how legal users can recover the messy phase data while preventing eavesdropping.* According to the previous study [7], the signals collected by the reader will be affected by the following three factors including power, frequency and hardware features. Since the phase is affected by various factors, it is difficult to be recovered through common data processing methods. A more challenging issue is that we have to ensure that the attacker cannot recover the phase data. In this paper, we use the authentication matrix to recover the phase data. We first calculate phase values under all given frequency and power conditions as the authentication matrix. Then, we choose a pair of power and frequency as references. Finally, we can uniformly map the phase value to the reference frequency and power through the derivation transformation of the phase equation, thus recovering the messy phase. It is worth noting that since the malicious attacker cannot obtain legal users' power and hardware information, it cannot exploit this matrix to recover phase data.

*The second challenge is how to reduce the impact caused by the frequency and power hopping delay.* When the reader switches transmission frequency and power, there will be a short stagnation. The shorter the interval is, the more

obvious the impact is. Such stagnation will affect the quality of dataset and reduce the accuracy of handwriting recognition. We use data smoothing and data augmentation to solve this challenge, we first use LOWESS [8] (locally-weighted scatter plot smoothing) to filter and smooth the data and then perform data augmentation. And with specific data augmentation methods, data collection costs for users are also greatly reduced.

The main contributions of this paper are as follows:

- We propose a privacy-preserving RFID handwriting recognition system called SecRFPen. The legal users can use our proposed phase recovery algorithm to enable correct handwriting recognition. But the malicious attacker cannot recognize the user's handwriting movement, because the eavesdropped tag phase profile seems messy and hard to understand.
- We employ a data augmentation method to improve dataset quality, thereby significantly reducing manpower during data collection. Also, a data smoothing method is used to reduce the impact caused by frequency and power hopping delay.
- We implement SecRFPen with COTS RFID devices. Extensive experimental results demonstrate that SecRFPen can protect user privacy without interfering handwriting recognition of legal users. The recognition accuracy of legal users is 94.2%, while the accuracy of malicious attacker is only 35.1%.

The remainder of this paper is organized as follows. We discuss the related works in Sect. 2. Section 3 introduces the preliminary knowledge. And in Sect. 4, we describe the attacking model. Section 5 presents the system design of SecRFPen. We discuss the implementation and experimental results in Sect. 6, and Sect. 7 concludes this paper.

## 2 Related Work

We discuss the following four categories of handwriting recognition works.

**Computer Vision-based Methods.** With the rapid development of machine learning, computer vision has become a common handwriting recognition scheme. Many studies [9,10] aim to improve the accuracy and robustness of the recognition process, and they use the feature information of pictures to classify. However, such methods also capture the user's face and body, which may easily lead to the leakage of user identity information, causing severe privacy concerns.

**Acoustic-based Methods.** The first acoustic handwriting recognition system Sonicnect [11] was proposed in 2016. Sonicnect takes advantage of the Amplitude Spectrum Density of the acoustic signal generated when handwriting and can recognize seven categories. Some systems [12,13] use acoustic signals in combination with well-designed deep learning frameworks for recognition. However, acoustic-based systems generally require an environment with lower ambient

noise, and the accuracy of recognition will decrease with writing distance. In addition, different writing strengths and pens will have a more significant impact on accuracy. In general, acoustic-based methods are less robust.

**Sensor-based Methods.** These systems recognize handwriting by capturing the features of the sensor when the user writes. For example, GyroPen [14] uses the general smartphone as a pen to record user movements with the gyroscopes and accelerometers that come with the phone. Pentelligence [15] combines acoustic signals with accelerators so that the two ways can complement each other's advantages. Unfortunately, the sensor also has some significant defects, such as a high failure rate and the need to replace the battery. Moreover, if the system uses the built-in sensor of the phone, the system also needs to access the necessary permissions of the mobile phone.

**RFID-based Methods.** There are some RF signal-based systems, such as RF-IDRAW [5], and the recent RF-copybook [6] and RF-Pen [4]. The principle of these systems is to collect the phase data of multiple tags for localization. RF-IDRAW uses a set of antenna arrays to collect phase data. RF-Copybook focuses on judging improvements in user writing behavior. RF-Pen can already achieve centimeter-level localization, and it uses both RSSI and phase data, the position of the tag is obtained by comprehensive voting. However, none of these system considers the privacy-preserving of user. The malicious attacker can eavesdrop in the RFID signals broadcast in the air and analyze the user's activity.

## 3 Preliminary

Our system focuses on RFID sensing systems using passive tags at the ultrahigh-frequency (UHF) because such systems are widely deployed in many sensing scenarios. The RFID reader can wake up the tag and report the readings in the RFID system, including Electronic Product Code (EPC), timestamp, phase, and RSSI. Among the information, the phase readings can reflect the fine-grained distance between the tag and the antenna because the COTS reader can support 0.00015 rad phase resolution [16]. In particular, the RFID phase $\theta_{i,j,t}$ can be expressed as follows:

$$\theta_{i,j,t} = \left[ \frac{2d_t}{\lambda_i} \cdot 2\pi + \theta_T(f_i, p_j) + \theta_R(f_i, p_j) + \theta_O \right] \mod 2\pi, \tag{1}$$

where $d_t$ is the distance between the antenna and the tag at time $t$, $f_i, p_j$ respectively represent the working frequency and power of the reader, $\lambda_i = c/f_i$ is the wavelength of RFID signal and $c$ is the speed of light. Consequently, the actual propagation distance of RF signals should be $2d_t$. Moreover, $\theta_T, \theta_R, \theta_O$ denote the phase offset by the tag's circuits, reader's internal circuit and relative orientation between antenna and tag.

## 4 Attacking Model

The malicious attacker aims to eavesdrop on the wireless signal in the air to recognize the user's handwriting. To exactly explain the threat model in our paper, we make the following assumptions:

**Passive Eavesdrop.** The malicious attacker can actively send wireless signals to request the tag response, but it is easy to find. The throughput of the reader is defined in Eq. (2).

$$throughput_{\Delta t} = \frac{N_{tags}}{Timestamp_{t+\Delta t} - Timestamp_t} \tag{2}$$

$N_{tags}$ indicates the number of times the tag is read, $\Delta t$ represents the time interval for reading. As statemented in C1G2 protocol, the reading rate of the tag is basically stable. When the throughput of legal readers suddenly decreases for a certain period, there is a high probability that an attacker exists. Therefore, the attacker uses passive eavesdropping method.

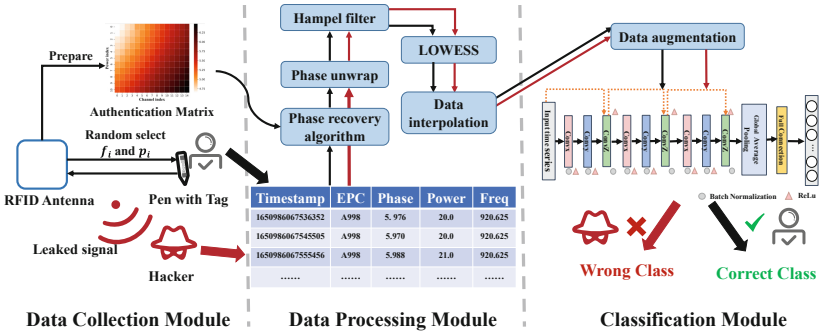



**Fig. 1.** System overview.

**One Antenna Only.** The attacker can implement eavesdropping in two ways. The first is the localization-based method. This method requires multiple antennas to locate the tag by calculating the intersection of the hyperbola or the maximum of the heat-map. Such methods have strict requirements on the distance between the antennas. The large distance can reduce the impact of phase errors on localization, such as in RF-PEN [4], the distance between antennas is $2\lambda$. However, such deployment greatly increases the exposure risk of the attacker. The other method is based on pattern recognition, which performs handwriting classification by analyzing the trend of the phase data, and only needs one antenna, which is convenient for concealment. Therefore we assume that the attacker only uses one antenna for eavesdropping.

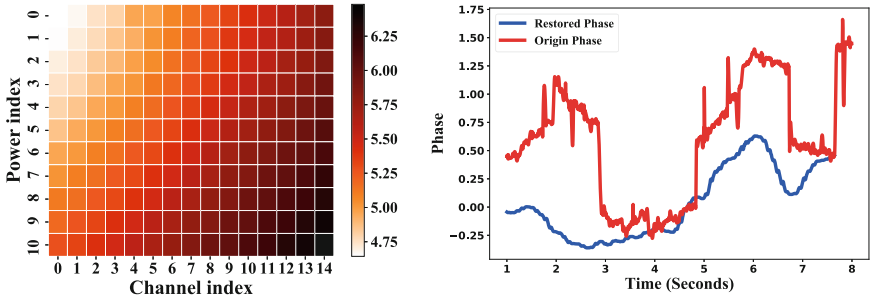
Based on the above assumptions, we give the following attacking model: The attacker can both acquire the wireless signal directly from the reader's antenna and the signal reflected by the tag. The phase reflected by the tag can be described as follows:

$$\theta'_{i,j,t} = \left[ \frac{(d_1 + d_2)}{\lambda_i} \cdot 2\pi + \theta_T(f_i, p_j) + \theta_R(f_i, p_j) + \theta_O + \theta_M \right] \mod 2\pi, \quad (3)$$

where $d_1$ represents the distance between the tag and the reader's antenna, $d_2$ represents the distance between the tag and the attacker, $\theta_M$ represents the phase offset by the attacker. Therefore, $\theta'_{i,j,t}$ can reflect the movement of the tag, and different movement methods will produce different patterns. The eavesdropper can realize the recognition of the user's handwriting through $\theta'_{i,j,t}$.

## 5 Detailed Design of SecRFPen

This section mainly introduces the design details of SecRFPen, including three modules: data collection, data preprocessing, and classification. Then, we will describe the design details from both perspectives of the malicious attacker and legal users, as shown in Fig. 1.




(a) Authentication matrix (b) Original and recovered phase

**Fig. 2.** Authentication matrix and phase contrast.

### 5.1 Data Collection

Before the system works, we first need to extract the authentication matrix, which is the premise of the phase recovery. Assuming that $M$ represents the number of candidate frequencies and $N$ is the number of candidate powers, we will get an $M \times N$ matrix, denoted by $\Theta$, where $\theta_{i,j}$ represents the phase reading collected at frequency $f_i$, power $p_j$.

Taking the ImpinJ R420 reader as an example, the frequency channel range is 920.625 MHz to 924.375 MHz with a step length of 0.25 MHz, and the power

range is 10 dBm to 31 dBm with a step length of 0.25 dBm. Considering real application scenarios, we collect phase readings with respect to all candidate frequency and some candidate power from 20 dBm to 30 dBm with a step length of 1 dBm. As shown in Fig. 2(a), the authentication matrix has strong regularity, we can find that the phase readings will increase as power and frequency increase. To summarize, the phase reading is positively correlated with power and frequency, and the phase difference under different frequencies and powers is noticeable. We test the phase matrices of various antennas and readers and find that the phase matrices obtained in each case are significantly different, which means that the attacker cannot obtain legal phase matrices.

When the system works, we utilize the RFID reader to receive the backscatter signal, and the system will record the EPC code of the tag, the current power of the reader, the frequency, phase readings, and time stamp as raw data. We let the frequency and power span about half of the total interval each time and stay in this condition for a short random time to collect the phase readings, this strategy can maximizes the phase difference obtained for each hopping.

### 5.2 Data Preprocessing

The original phase data readings are messy and we need to recover the data so that the readings can be correctly classified. Inspired by the work of Wei [17] and according to Eq. (1), we further add the effect of the power factor. By switching frequency and power simultaneously, we are able to increase the complexity of the authentication matrix, which makes it more difficult for the malicious attacker to recover the original phase data.

Equation (4) and Eq. (5) represent the phase when we acquire the authentication matrix, $d_0$ represents the distance when we collect the authentication matrix, which does not need to be measured in advance.

$$\theta_{i,j,d_0} = \left[\frac{2d_0}{\lambda_i} \cdot 2\pi + \theta_T(f_i, p_j) + \theta_R(f_i, p_j) + \theta_O\right] \mod 2\pi \tag{4}$$

$$\theta_{r,r,d_0} = \left[\frac{2d_0}{\lambda_i} \cdot 2\pi + \theta_T(f_r, p_r) + \theta_R(f_r, p_r) + \theta_O\right] \mod 2\pi \tag{5}$$

$$\theta_{i,j,d} = \left[\frac{2d}{\lambda_i} \cdot 2\pi + \theta_T(f_i, p_j) + \theta_R(f_i, p_j) + \theta_O\right] \mod 2\pi \tag{6}$$

$$\theta_{r,r,d} = \left[\frac{2d}{\lambda_i} \cdot 2\pi + \theta_T(f_r, p_r) + \theta_R(f_r, p_r) + \theta_O\right] \mod 2\pi \tag{7}$$

We pre-select a reference frequency $f_r$ and a reference power $p_r$. During the collecting process, the phase readings at different frequencies and powers are uniformly converted to $f_r$ and $p_r$. Equation (6) represents the phase readings in the actual acquisition, and Eq. (7) represents the converted phase readings. After derivation, we can derive Eq. (8). According to this equation, we only need to obtain the authentication matrix in advance, so that we are able to recover the messy phase sequence into clear phase sequence, as shown in Fig. 2(b).

$$\theta_{r,r,d} = (\theta_{i,j,d} - \theta_{i,j,d_0}) \times \frac{f_r}{f_i} + \theta_{r,r,d_0} \tag{8}$$

The malicious attacker can not obtain the legitimate authentication matrix in advance, and it can be seen from Eq. 1 that different tags, readers, and antennas will change the phase reading significantly. Therefore, even if the malicious attacker obtains the authentication matrix through other RFID systems, the matrix is quite different from our legitimate authentication matrix.

During handwriting, as the position of the tag might change by more than half a wavelength, the phase readings would jump. Hence, we need to carry out the unwrap operation to recover the phase. By unwrapping, the discrete pieces of the phase turn into consecutive phase data. We extend the public function in Numpy as the unwrap function used by our system.

After unwrapping, the data needs to be further filtered. We use two methods. The first method is Hampel filtering, which detects and removes outliers in the data. Given a window value $K$, calculate the median value $m_i$ of all elements in the window, and use the $|m_i|$ to estimate the standard deviation of the samples, denoted as $\sigma_i$. If the interpolation of $x_i$ and $m_i$ exceeds $3\sigma_i$, use $m_i$ to replace the sample, $x_i$ represents the $i_{th}$ phase value under the window $K$.

The second method is Lowess, called locally weighted scatter plot smoothly. The general idea of Lowess is to take a point $x$ as the center and intercept a piece of data with a $frac$ forward and backward. We use the weight function $w$ to perform a weighted linear regression for this piece of data. We denote $(x, \hat{y})$ as the center value of the regression line, where $\hat{y}$ is the corresponding value of the fitted curve. For $n$ data points, $n$ weighted regression lines can be made, and the center of each regression line is connected to the Lowess curve of the data. We choose the Eq. (9) as the weight function $w$, where $x$ is the sample to be predicted, $x_i$ is the sample around $x$, and $\tau$ is the decay factor, we empirically set the value of $\tau$ to 1. Finally, we interpolate the data to reduce the sampling rate and make the dataset smaller for training and classification.

$$w_i = exp(-\frac{(x_i - x)^2}{2\tau^2}) \tag{9}$$

### 5.3 Classification Module

In order to address the problems caused by poor data quality and time-consuming data collection, we need to perform data augmentation to expand the dataset.

For this purpose, we first adopt the traditional time series data augmentation [18] method. We apply the window slicing on the entire sequence data, where we randomly crop a fixed proportion of the sequence. Window wrapping is then applied, so that we can stretch or shrink randomly the selected sub-windows. After the above two-step processing, the size of data will become 15 times the original length. Next, we further process all the data by adding Gaussian noise. And we choose two of the most effective methods in Um's work [18], magic wrap

and time wrap. The magic wrap makes each phase reading scale slightly. In contrast, time wrap scales the timestamp reading to a certain extent and then re-interpolates the sampling. Both methods make the data more diverse.

We also implement a DBA (DTW Barycentgric Avereaging) data augmentation method. First, we randomly select an initial phase sequence $X$ from the dataset, give $X$ a weight of 0.5, and find the five closest phase sequence according to the DTW distance. Then we randomly select two of these five sequences and give a weight of 0.15 respectively. The remaining sequences are equally divided into a weight of 0.2, and a new sequence $X_{gene}$ is finally generated. This method is an efficient and common method in time series augmentation.

In the classification part, we choose ResNet (Residual neural network).

## 6 Performance Evaluation

In this section, we perform extensive experiments to evaluate the performance of SecRFPen. We first describe the system implementation and experimental setup and then evaluate the system performance under multiple conditions.

### 6.1 Implementation

We describe our implementation in terms of both hardware and software.

**Hardware:** System hardware includes an ImpinJ Speedway R420 RFID reader, Alien AZ-9640 tags, and Laird S9028PCR RFID antennas. Our algorithm is implemented in a ThinkPad laptop equipped with an Intel i7-8550U CPU and 8GB RAM. We connect reader and laptop via ASUS Wireless USB Adapter for long-distance connection. In this experiment, We stick the tags to pens, the antenna is placed parallel to the front of the pen, The overall experimental deployment is shown in Fig. 3.

In the classification module, we adopt the ResNet network architecture of 3 Blocks, The size of the feature map in the first block is 64, and the other feature map's size is 128. A Batch Normalization layer follows each convolution layer. Finally, it is output to the fully connected layer through an average pooling layer, and we choose the cross-entropy as the loss function. After many experiments, we find that the optimal network learning rate is 0.001 and the batch size is 16. We divide the collected data into two parts: the test set and the training set, and we perform data augmentation on the training set.

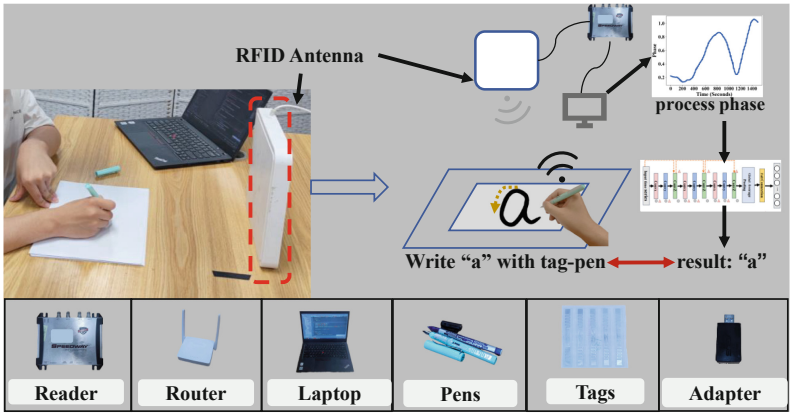
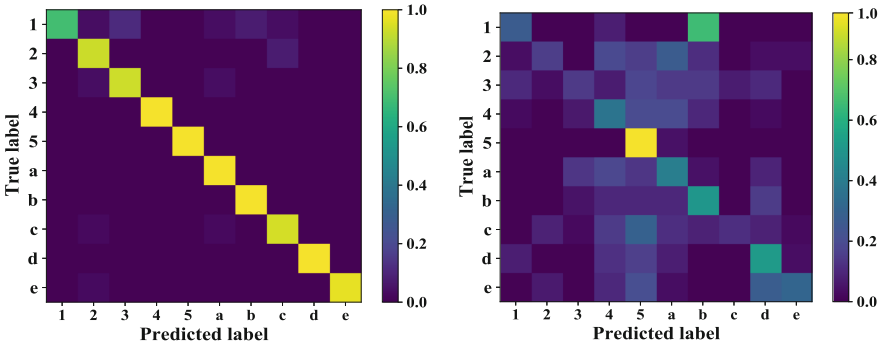



**Fig. 3.** The schematic diagram of the experiment and hardware devices.




(a) Legal user (b) Malicious attacker

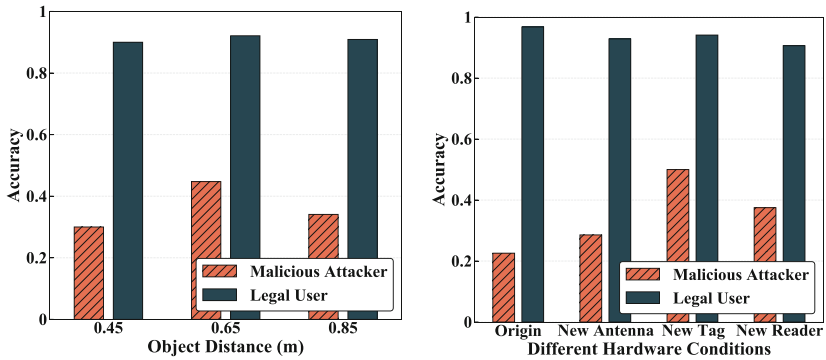
**Fig. 4.** Confusion matrix for legal user and malicious attacker.

The Malicious attacker also uses DNN model to recognize handwriting. The model is trained with the data collected under stable transmission frequency and power. The attacker can also take transfer learning to get the model, but both methods must assume the tag position. However, since the relative position between tag and attacker is unknown during the actual scene, it is of great possibility that the trained model utilized by the attacker performs poorly. For the sake of the Malicious attacker, we treat the data not processed by the recovery algorithm as the data the attacker can collect. This should be the best phase data that the malicious attacker can have.

### 6.2 Accuracy of Handwriting Recognition

In the experiment, we invited a total of 5 volunteers, including three men and two women to help us evaluate the accuracy of the system, and we test a total

of five letters $(a, b, c, d, e)$ and five numbers *1,2,3,4,5*, ten characters in total. In order to increase the robustness of the system, we set up multiple sets of different conditions and let volunteers write each character 20 times under multiple conditions. The classification accuracy is shown in Fig. 4. The experimental results show that the recognition accuracy of legal users is 94.2%, while the recognition accuracy of the malicious attacker is only 35.1%, there is a big gap between the two.




(a) Impact of distance (b) Impact of hardware

**Fig. 5.** Investigating the impact of distance and hardware.

Figure 4(a) is the confusion matrix of legal users. We can see that our system can recognize and classify all characters well. Among the ten characters, the recognition accuracy of *1,2,3,c* is slightly lower than other characters. This is because the phase profile of these characters is relatively similar, and their overall profile is rising. Figure 4(b) is the confusion matrix of the malicious attacker. It is worth mentioned that the malicious attacker's data is also filtered and unwrapped, but the phase recovery algorithm is not applied. Therefore, it is the phase readings that the malicious attacker can eavesdrop in the ideal environment. It can be seen from the confusion matrix that there is no clear rule in the classification of malicious attacker. Only the character *5* has a higher recognition accuracy, which may be related to the complicated writing trend of *5*.

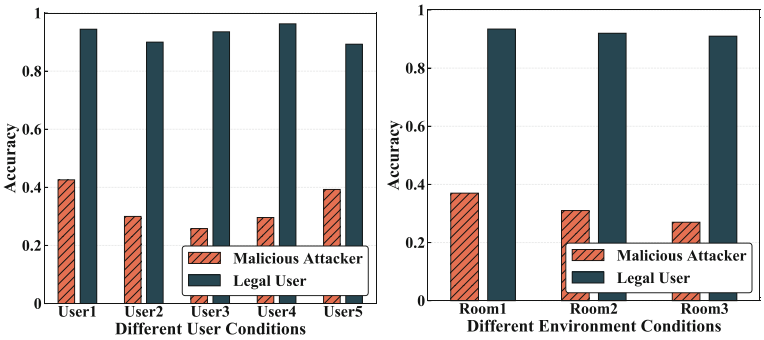
### 6.3 Investigating the Impact of Different Conditions

We consider the influence of four conditions including distance, hardware, user, and system environment. In addition, we ensure that other conditions are the same when verifying a specific condition. Finally, we will present the experimental results of both legal users and malicious attackers.

Figure 5(a) shows the accuracy at different distances. When investigating the distance condition, we set three conditions of 0.45 $m$, 0.65 m, and 0.85 m. The farther distance does not meet the premise of handwriting recognition. It

can be seen that the system is less affected by the distance conditions, and the proposed system is able to maintain a high accuracy rate at different distances. The malicious attacker's accuracy is slightly higher at 0.65 $m$.

Figure 5(b) is the accuracy under different hardware. When verifying the hardware conditions, we select different antennas, tags, and readers according to the Eq. (1). We ensure that the readers and tags remain unchanged when investigating the antenna condition, and the rest are the same. Since our certification matrix is dependent on hardware characteristics, we need to update the certification matrix in time when considering hardware conditions, and each condition requires a unique certification matrix. It can be seen that the hardware conditions will not cause significant fluctuations in the accuracy. In these conditions, the recognition accuracy of the malicious attacker is high after changing the new label, while the recognition accuracy of the legal user remains stable and high.




(a) Impact of user (b) Impact of environment

**Fig. 6.** Investigating the impact of user and environment.

Figure 6(a) shows the accuracy of writing by different users. Because people's writing styles, speed, and size are different, in order to avoid the impact on user characteristics, we prepare a copybook in advance to ensure that the size of the characters is the same. We invite a total of 5 volunteers, there are certain differences in the accuracy of different users, but the difference is not significant, and they are all maintained at more than 90%, which to a certain extent also shows the robustness of our system.

We also take into account the impact of the environment, we invite a volunteer to write in three different rooms, and the result is shown in Fig. 6(b). It can be seen that the environmental conditions have a negligible effect on the accuracy. We consider that the tag and the reader are close to each other and will not be affected by multi-path effects.

## 7 Conclusion

In this paper, we proposed a privacy-preserving handwriting recognition system named SecRFPen by hopping frequency and power. SecRFPen can guarantee the high accuracy of handwriting recognition while preventing the malicious attacker from recognizing user's handwriting activities. Specifically, We first collected a set of phase data at each power and frequency to form the authentication matrix. With this authentication matrix and our proposed recovery algorithm, legal users can easily get precise phase data, but the malicious attacker cannot. Our system can seamlessly be deployed on COTS RFID devices without additional implementation costs. The experimental results show that SecRFPen can enable secure RFID handwriting recognition by ensuring high accuracy for legal users and low accuracy for malicious attackers.

**Acknowledgment.** This work is supported in part by the National Natural Science Foundation of China under Grant Nos. 62002259, 62032017.

## References

1. Alam, M.S., Kwon, K., Kim, N.: Implementation of a character recognition system based on finger-joint tracking using a depth camera. IEEE Trans. Hum. Mach. Syst. **51**(3), 229–241 (2021)
2. Ding, D., Yang, L., Chen, Y., Xue,G.: VibWriter: handwriting recognition system based on vibration signal. In: Proceedings of IEEE SECON, pp. 1–9 (2021)
3. Chen, M., AlRegib, G., Juang, B.-H.: Air-writing recognition–part I: modeling and recognition of characters, words, and connecting motions. IEEE Trans. Hum. Mach. Syst. **46**(3), 403–413 (2015)
4. Wang, H., Gong, W.: RF-pen: practical real-time RFID tracking in the air. IEEE Trans. Mobile Comput. **20**(11), 3227–3238 (2020)
5. Wang, J., Vasisht, D., Katabi, D.: RF-IDraw: virtual touch screen in the air using RF signals. vol. 44, pp. 235–246 (2014)
6. Chang, L., Jie Xiong, J., Wang, X.C., Wang, Yu., Tang, Z., Fang, D.: RF-copybook: A millimeter level calligraphy copybook based on commodity RFID. Proc. ACM on Interact. Mob. Wearable Ubiquit. Technol. **1**(4), 1–19 (2018)
7. Jiang, C., He, Y., Zheng, X., Liu, Y.: OmniTrack: orientation-aware RFID tracking with centimeter-level accuracy. IEEE Trans. Mob. Comput. **20**(2), 634–646 (2019)
8. Cleveland, W.S.: Lowess: a program for smoothing scatterplots by robust locally weighted regression. Am. Stat. **35**(1), 54 (1981)
9. Xiao, X., Jin, L., Yang, Y., Yang, W., Sun, J., Chang, T.: Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition. Pattern Recognit. **72**, 72–81 (2017)
10. Poznanski, A., Wolf, L.: CNN-N-Gram for handwriting word recognition. In: Proceedings of IEEE CVPR, pp. 2305–2314 (2016)
11. Zhang, M., Li, P., Yang, P., Xiong, J., Chang, T.: Poster: sonicnect: accurate hands-free gesture input system with smart acoustic sensing. In: Proceedings of ACM MobiSys, pp. 91–91 (2016)
12. Du, H., Li, P., Zhou, H., Gong, W., Luo, G., Yang, P.: Wordrecorder: accurate acoustic-based handwriting recognition using deep learning. In: Proceedings of IEEE INFOCOM, pp. 1448–1456 (2018)

13. Kaishun, W., Yang, Q., Yuan, B., Zou, Y., Ruby, R., Li, M.: Echowrite: an acoustic-based finger input system without training. IEEE Trans. Mob. Comput. **20**(5), 1789–1803 (2020)
14. Deselaers, T., Keysers, D., Hosang, J., Rowley, H.A.: Gyropen: gyroscopes for pen-input with mobile phones. IEEE Trans. Hum. Mach. Syst. **45**(2), 263–271 (2014)
15. Schrapel, M., Stadler, M.L., Rohs, M.: Pentelligence: combining pen tip motion and writing sounds for handwritten digit recognition. In: Proceedings of ACM CHI, pp. 1–11 (2018)
16. Yang, L., Chen, Y., Li, X.-Y., Xiao, C., Li, M., Liu, Y.: Tagoram: real-time tracking of mobile RFID tags to high precision using COTS devices. In: Proceedings of ACM MobiCom, pp. 237–248 (2014)
17. Wei, T., Zhang, X.: Gyro in the air: tracking 3D orientation of batteryless internet-of-things. In: Proceedings of ACM MobiCom, pp. 55–68 (2016)
18. Um, T.T., et al.: Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks. In: Proceedings of ACM ICMI, pp. 216–220 (2017)

# Privacy Preserving Federated Learning Using CKKS Homomorphic Encryption

Fengyuan Qiu[1], Hao Yang[1], Lu Zhou[1(✉)], Chuan Ma[2], and LiMing Fang[1,3]

[1] Nanjing University of Aeronautics and Astronautics, Jiangsu, China
lu.zhou@nuaa.edu.cn
[2] Nanjing University of Science and Technology, Jiangsu, China
[3] Nanjing University of Aeronautics and Astronautics Shenzhen Research Institute, Guangdong, China

**Abstract.** With the rapid development of distributed machine learning and Internet of things, tons of distributed data created by devices are used for model training and what comes along is the concern of security and privacy. Traditional method of distributed machine learning asks devices to upload their raw data to a server, which may cause the privacy leakage. Federated learning mitigates this problem by sharing each devices' model parameters only. However, it still has the risk of privacy leakage due to the weak security of model parameters. In this paper, we propose a scheme called privacy enhanced federated averaging (PE-FedAvg) to enhance the security of model parameters. By the way, our scheme achieves the same training effect as Fedavg do at the cost of extra but acceptable time and has better performances on communication and computation cost compared with Paillier based federated averaging. The scheme uses the CKKS homomorphic encryption to encrypt the model parameters, provided by detailed scheme design and security analysis. To verify the effectiveness of the proposed algorithm, extensive experiments are conducted in two real-life datasets, and shows the advantages on aspects of communication and computation. Finally, we discuss the feasibility of deployment on IoT devices.

**Keywords:** Homomorphic encryption · Federated learning · Privacy preserving · IoT

## 1 Introduction

It is estimated that the global IoT devices will reach 75 billion by 2025 [20]. These devices are widely deployed around our lives, which create tons of distributed data, and these data can make a big effect in such scenarios like smart health, smart city and smart traffic. Data has become a new type of treasure in information era today. However, most of the data generated by IoT devices is private such as personal health records, travel activities, loan records and so on. These data may be disclosed during the process of training model which brings


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 427–440, 2022.
https://doi.org/10.1007/978-3-031-19208-1_35

a big threat to personal privacy protection. Fortunately, Google has proposed a scheme called Federated Learning (FL) [13] to solve this problem properly.

FL is a special type of distributed machine learning. It allows participants with different data to train model together and in the whole process of training, each participant only need to share their local training results, the model parameters, which usually are gradients. Through this way, we can let each participant hold their own data in their devices which avoids uploading private data to a data processing center. Obviously, FL not only can protect privacy of participants, but also solves the problem of high communication cost by sharing model parameters. The training process is shown in Fig. 1.

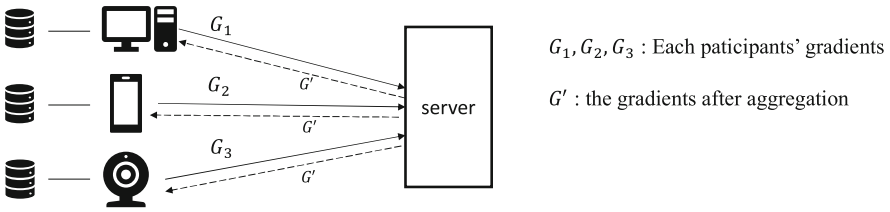



**Fig. 1.** The training process of Federated Learning.

Unluckily, FL still faces many threats in privacy protection. One of the threats is gradients attack. Hitaj et al. [12] proposed a kind of gradients attack based on generative adversarial network, which can restore any participants' training dataset by analyzing their gradients. Song et al. [17,19], proposed the ideas of membership inference attack and attribute inference attack. The former one can be used to judge whether a data point is in someone's training dataset and the latter one can be used to judge whether someone's training dataset contains certain attributes. Besides, Zhu et al. [23] proposed a method called deep leakage gradients, it can reconstruct the real training datasets by adjusting fake training datasets and computing the loss between real and fake gradients. However, there are already many way to handle this problem properly, one of them is homomorphic encryption (HE).

HE is a special encryption, which can let results computed on unencrypted data equals to the decrypted results computed on encrypted data. So, we transmit data encrypted by HE to an untrusted third party and let the party execute computation tasks and send encrypted results back to us. In the whole process shown as Fig. 2, the third party knows nothing from ciphertext. In this way, we can protect our privacy away from gradients attack.

In this paper, we propose a scheme called privacy enhances federated averaging (PE-FedAvg) based on FedAvg and CKKS. To verify the effectiveness, we deploy our scheme on two IoT devices and one server to train a model and analyze the communication cost and training effect. Our detailed contributions are as follows:

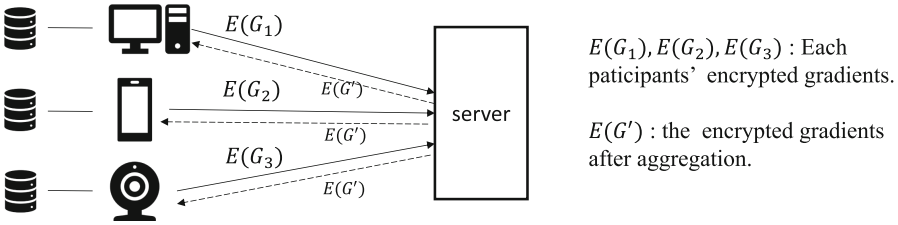



**Fig. 2.** the training process of Privacy Enhanced Federated Learning.

1. We propose a scheme called Privacy Enhanced Federated Averaging (PE-FedAvg) which can protect participant's private data away from gradients attack. The scheme uses homomorphic encryption to encrypt model parameters so that the server infer nothing from ciphertext through gradients attack and only finish its aggregation job.
2. We use the CKKS homomorphic encryption to encrypt model parameters which has lower expansion multiple in ciphertext and faster computing performance compared with Paillier homomorphic encryption. What's more, our scheme is based on FedAvg, which can relieve our scheme's communication cost in some point caused by HE. These techniques make our scheme possible to be deployed in large scale federated learning scenarios.
3. To verify the effectiveness of the proposed algorithm, extensive experiments are conducted in two real-life datasets, and shows the advantages on aspects of communication and computation.

## 2 Related Work

Most of our work is related to prior researches on HE and FL, in particular, with respect to privacy preserving federated learning schemes.

### 2.1 Privacy Preserving Federated Learning

Up to now, there have been many privacy preservation methods for gradients, which can be roughly divided into three following categories: cryptography, differential privacy and gradients compression, each with its own advantages and disadvantages.

In all of strategies of defensing gradients attack, the category of cryptography is the safest one theoretically. Here are two methods of it which is very popular, one is homomorphic encryption and we will discuss it later. The another one is security multi-party computation (SMPC). Bonawitz et al. [3,4] applied the scheme of SMPC to federated learning so that participants can share their model parameters safely without concerning the leakage of their own model. However, challenges still exist such as most of SMPC schemes are complicated to be used and the negotiation before sharing parameters usually brings the

problem of explosive communication cost. Compared with SMPC schemes, HE schemes usually bring a relatively smaller communication cost and are easier to be used. Even though, homomorphic encryption still brings a significant communication cost.

Another category of defensing gradients attack is differential privacy [9]. Nowadays, tremendous researches and applications related it have come out since it was introduced to people. One of them is the application on defensing gradients attack [22]. Each participant add noise to gradients before sharing them to server and the noise usually is Gauss noise or Laplacian noise. So when attacker want to restore data from gradients, what they get is fuzzy data. However, the strategy merely reduces the leakage of information, besides, the defensing effect is related to the amount of noises added to the gradients, which will degrade the learning performance.

The last category is gradients compression. It is a rather intuitive way to solve this problem. Each participant compresses their gradients before sending them to server, so that it will be harder to restore from it. Lin et al. [14] first proposed this idea to reduce communication bandwidth and some researches [14,21] showed that gradients can be compressed by more than 300 times without loss of accuracy. Although it is a good idea, it is only suitable on big model and cannot defense the gradients totally.

### 2.2 Federated Learning with Homomorphic Encryption

Aono et al. [1] proposed a scheme of privacy preserving deep machine learning using LWE-based additive homomorphic encryption on gradients and theoretically analyse the expansion multiple. However the scheme does not give a solution to solve the problem of high communication cost. Fang and Qian [10] use the Paillier homormorphic encryption [18] on federated learning but Paillier scheme will bring large communication cost when the encrypted object's number is big enough and the computation is not efficient compared with CKKS scheme [7]. Besides, the Paillier scheme does not have that characteristic of anti-quantum.

## 3 Preliminaries

The scheme PE-FedAvg involves federated learning, homomorphic encryption and some details like FedAvg, CKKS. We introduce these knowledges briefly in this section.

### 3.1 Homomorphic Encryption

Homomorphic encryption is not only a kind of encryption method but also a kind of special computation method. It allows computing directly on ciphertext and after decryption, the results will be the same as computing directly on plaintext. Usually, the computation is addition or multiplication. Let us define encryption

function $E$, decryption function $D$ and plaintext $a$,$b$.The concept can be shown as follows:

$$D(E(a) \odot E(b)) = a \times b \tag{1}$$

$$D(E(a) \oplus (b)) = a + b \tag{2}$$

The scheme is called partial homomorphic encryption (PHE) if it only holds for (1) or (2) and is called full homomorphic encryption (FHE) if it satisfies both. Specially, it is called level homomorphic encryption (LHE) if it only supports limited times of computation.

**CKKS.** CKKS [7] is LHE. It supports addition and multiplication but the times of computation is limited. However, with the use of bootstrapping technique [5,6], the CKKS scheme becomes a FHE. The decrypted results are approximate number compared to plaintext, which means it is acceptable for machining learning. Besides, it is an encryption based on lattice which brings it a new characteristic, anti-quantum.

Let $N = \phi(M)$ be the degree of the $M$-$th$ cyclotomic polynomial $\Phi_M(X)$. If $N$ is chosen as a power of 2, then $M = 2N$, and the $M$-$th$ cyclotomic polynomial $\Phi_M(X) = X^N + 1$. Let $R = Z[X]/\Phi_M(X) = Z[X]/(X^N + 1)$ be the ring of polynomials defined for the plaintext space. Let $R_q = R/qR = Z_q[X]/(X^N + 1)$ be the residue ring defined for the ciphertext space. Let $H$ be a subspace of $C^N$, which is isomorphic to $C^{N/2}$. Let $\sigma : R \to \sigma(R) \subseteq H$ be a canonical embedding. Let $\pi : H \to C^{N/2}$ be a map that projects a vector from a subspace of $C^N$ to $C^{N/2}$. The CKKS scheme mainly provides the following operations:

- **KeyGen($N$)** Let $s(X) \in Z_q[X]/(X^N + 1)$ be the secret polynomial and $p(X) = (-a(X) \cdot s(X) + e(X), a(X))$ be the public polynomial where $a(X) \in Z_q[X]/(X^N + 1)$ is a polynomial chosen uniformly random and $e(X) \in Z_q[X]/(X^N + 1)$ is a small noisy polynomial. Let $r(x) = (-a(X) \cdot s(X) + b \cdot s(X)^2 + e(X), a(X))$ be the relinearisation key where $b \in Z_q$ is a large integer.
- **Encode($z$)** To encode a message vector $z \in C^{N/2}$ to a message polynomial $m(X) \in R$ , we first expand the message vector $z$ from $C^{N/2}$ to $H$ by applying $\pi^{-1}(z)$. Then we appropriately scale the vector by multiplying a scaling factor. $\Delta$ followed by random rounding to $\lfloor \Delta \cdot \pi^{-1}(z) \rceil$. Scaling is done to achieve predefined precision since precision bits may be lost due to rounding. To obtain the message polynomial, we apply the inverse of canonical embedding $\sigma^{-1}$ and get $m(X) = \sigma^{-1}(\lfloor \Delta \cdot \pi^{-1}(z) \rceil) \in R$.
- **Decode($m(X)$)** To decode a message polynomial $m(X) \in R$ to a message vector $z \in C^{N/2}$, we first apply the canonical embedding $\sigma$ to get $z = \lfloor \Delta \cdot \pi^{-1}(z) \rceil \in H$. Then, we divide it by the scaling factor $\Delta$ to obtain $\Delta^{-1} \lfloor \Delta \cdot \pi^{-1}(z) \rceil \approx \pi^{-1}(z)$. To obtain the message vector, we project the vector using $\pi$ and get $\pi(\pi^{-1}(z)) = z \in C^{N/2}$.
- **Encrypt($m(X), p(X)$)** To obtain the ciphertext polynomial $c(X)$ corresponding to the message polynomial $m(X) \in R$, we apply the RLWE encryption and get $c(X) = (c_0(X), c_1(X)) = (m(X), 0) + p(X) = (m(X) - a(X) \cdot s(X) + e(X), a(X)) \in (Z_q[X]/(X^N + 1))^2$.

– **Decrypt**$(c(X), s(X))$ To obtain the message polynomial corresponding to the ciphertext polynomial $c(X) \in (Z_q[X]/(X^N + 1))$, we apply the RLWE decryption using the secret polynomial $s(X)$ and get $m(X) \approx c_0(X) + c_1(X) \cdot s = m(X) + e(X)$.

### 3.2 Federated Learning

The classic federated learning is centralized. Each participant sends their own model parameters to a server and the server executes the operation of aggregation and sends the results back to each participant. The process can be described as follow:

1. Each participants $P_i$ downloads initial parameters $w_t$ from server.
2. Participant $P_i$ update its model parameters to $w_t$ and train its model on local data to get new parameters $w_{t+1}^i$. After that, sends them to server.
3. The server will do the aggregation after receiving all participants' model parameters and get the result $w_{t+1}$. After that, it sends $w_{t+1}$ back to each participant.
4. Repeat the step 2 and step 3 until convergence.

**FedAvg.** Federated averaging (FedAvg) is one of many algorithms of federated learning. It was first proposed by McMahan et al. [16] to reduce the communication cost and shrink down the training time. What's the main difference compared with classic federated learning is that not all participants take part in training in one round. It chooses some participants randomly each round and researches show that the effect of training is nice. Suppose we have devices $d_i$, $i \in 1, \ldots, N$ and we will choose $N'$ devices each round to train our model. In round $t + 1$, device $d_i$ chosen trains their model on parameters $w_t$ using local data set and get new one $w_{t+1}^i$ which then is sent to server. The server sums all the parameters of chosen devices and averages them to get the new parameters $w_{t+1}$, and that is the end of one round training. The pseudocode of the algorithm is shown as Algorithm 1.

## 4 Privacy Enhanced Federated Averaging

In this section, we will provide detailed design and algorithm of our scheme. The key idea is using the algorithm of FedAvg to reduce the communication cost and using homomorphic encryption based on lattice to encrypt model parameters.

### 4.1 Design of PE-FedAvg

The code of federated averaging references from FedML library [11] which is feasible to be deployed on IoT scenario and the CKKS scheme we use comes from TenSEAL library [2] which provides a list of easy and secure API for developer. We refactor the code of federated averaging and use CKKS provided by TenSEAL to encrypt the model parameters. Let us suppose that the ciphertext is $E(m)$, $m$ is plaintext, $w$ is the model parameters and the algorithm flow is as follows:

**Algorithm 1** Federated Averaging (FedAvg)

Let $\eta$ be the learning rate, $B$ be the minibatch size for local model training and $n = \sum_{i=1}^{N'} n_i$, $n_i$, which is size of dataset in device $d_i$.
The server initializes $w_0$.
//server
**for** each round $t = 1, 2, \ldots$ **do**
  $S_t \leftarrow$ (server randomly selects a set of devices $d_i$ sized of $N'$)
  **for** each remote device $d_i \in S_t$ in parallel **do**
    $w_{t+1}^i \leftarrow DeviceUpdate\,(d_i, w_t)$
  **end for**
  $w_{t+1} \leftarrow \frac{1}{N'} \sum_{i=1}^{N'} \frac{n_i}{n} w_{t+1}^i$
  **if** Satisfy termination condition **then**
    break
  **end if**
**end for**
//device
$DeviceUpdate\left(d_i, w_t^i\right)$ :
$S \leftarrow$(select batches sized of $B$ from local dataset)
//we use batch stochastic gradients descent to iterate model
**for** local epochs $l = 1, \ldots, L$ **do**
  **for** batch $b \in S$ **do**
    $//\nabla loss$ is the gradients of loss function.
    $w_{t+1}^i \leftarrow w_t^i - \eta \nabla loss\left(w_t^i; b\right)$
  **end for**
**end for**
return $w_{t+1}^i$ to server

**Setup.** Key server or one special participant initializes the CKKS parameters and generates public key $pk$ and secret key $sk$, after that, sends the public key and secret key to all participants and sends public key to data process server. The data process server then initializes model parameters $w_t$.

**Step 1 Selecting Participants.** Server selects a certain number $N'$ of participants from all participants and if the participants' num is smaller than $N'$, selects all. The participant $P_i$ selected by server then receives model parameters $w_t$ or $E\,(w_t)$. The former is initial parameters and the latter one is ciphertext during the training process.

**Step 2 Local Training.** After receiving parameters from server, participant $P_i$ judge the parameters first whether is ciphertext, if yes then decrypts the ciphertext first using secret key $sk$. Then $P_i$ updates their own model using the new parameters and trains model to get a better parameters $w_{t+1}^i$.

**Step 3 Encryption.** Participants need to encrypt the new parameters $w_{t+1}^i$ using the public key $pk$ and then send $E\left(w_{t+1}^i\right)$ back to server.

**Step 4 Aggregation.** The server aggregates all the parameters received from selected participants and get new parameters $E(w_{t+1})$. Specially, all the parameters server holds are ciphertext, the server know nothing from it. The equation shown below presents the process of aggregation.

$$E(w_{t+1}) = Avg\left(E\left(w_{t+1}^{1}\right), \ldots, E\left(w_{t+1}^{N'}\right), pk\right) \tag{3}$$

**Step 5 Loop.** Repeating the Step 1 to Step 4 until the model converges or other termination conditions are satisfied.

To further understand PE-FedAvg, we provide the pseudocode shown as Algorithm 2.

**Algorithm 2** Privacy Enhanced Federated Averaging (PE-FedAvg)

Let $\eta$ be the learning rate, $B$ be the minibatch size for local model training and $n = \sum_{i=1}^{N'} n_i$, $n_i$ is size of dataset in device $d_i$.
Key server initializes the CKKS parameters, generates the public key $pk$ and secret key $sk$ and sends $sk$, $pk$ to all participants, sends $pk$ to server only.
The server initializes $w_0$.
//server
**for** each round $t = 1, 2, \ldots$ **do**
  $S_t \leftarrow$ (server randomly selects a set of devices $d_i$ sized of $N'$)
  **for** each remote device $d_i \in S_t$ in parallel **do**
    $E(w_{t+1}^{i}) \leftarrow DeviceUpdate(d_i, w_t)$
  **end for**
  $E(w_{t+1}) \leftarrow \frac{1}{N'} \sum_{i=1}^{N'} \frac{n_i}{n} E(w_{t+1}^{i})$
  **if** Satisfy termination condition **then**
    break
  **end if**
**end for**
//device
$DeviceUpdate\left(d_i, E(w_t^i)\right)$ :
$w_t^i \leftarrow Decrypt\left(E\left(w_t\right), sk\right)$
$S \leftarrow$(select batches sized of $B$ from local dataset)
//we use batch stochastic gradients descent to iterate model
**for** local epochs $l = 1, \ldots, L$ **do**
  **for** batch $b \in S$ **do**
    //$\nabla loss$ is the gradients of loss function.
    $w_{t+1}^{i} \leftarrow w_t^i - \eta \nabla loss\left(w_t^i; b\right)$
  **end for**
**end for**
$E\left(w_{t+1}^{i}\right) \leftarrow Encrypt\left(w_{t+1}^{i}, pk\right)$
return $E(w_{t+1}^{i})$ to server

### 4.2 Security Analysis

In this section, we are going to discuss how our scheme can guarantee the data privacy of each participant and ensure the confidentiality of model parameters.

We use a common assumption that the server is honest but curious, and in this setting, the server may infer the participant's information from the uploaded models. In our scheme, each participants trains their own model locally and encrypts the model parameters by CKKS before sending them to server. So the server receives ciphertext from each participants and the only thing is to execute computation tasks on these ciphertext. The final results after aggregation are also ciphertext and will be sent to the participants. After that, they will be decrypted and used to train local model in next round. During the whole process, the server can infer nothing from ciphertext except cracking it. Besides, the existing quantum attack algorithm is aimed at integer decomposition and the lattice-dependent quantum attack has not yet come out. That's why the CKKS, lattice-based homomorphic encryption, have the characteristic of anti-quantum.

In short, our scheme has the security against honest but curious server and also has anti-quantum characteristic in a way.

### 4.3 Communication Cost Analysis

It's worth mentioning that our scheme use the function $ckks\_vector()$ of TenSEAL and so the expansion multiple of a matrix, sized of $N \times M$, is depended on $M$ not $NM$ and the encrypted object's datatype is list. Let us assume a neural network's weights matrix is $N \times M$ and the bias is $1 \times N$, $f(x)$ is a function to map the list's size $x$ to corresponding expansion multiple, $size(x)$ is a function to compute the memory space of a list sized of $x$ and $l$ is the communication cost except the model parameters' cost in one round which is a constant. In this way, we can obtain the mathematical expression of the biggest expansion multiple $T_{\text{biggest}}$ of the communication cost of PE-FedAvg scheme which is represented as follows:

$$\frac{size\left(NM+N\right)\left(\frac{NM}{NM+N}f\left(M\right)+\frac{N}{NM+N}f\left(N\right)\right)+l}{size\left(NM+N\right)+l}$$

$$\leq \frac{NM}{NM+N}f\left(M\right)+\frac{N}{NM+N}f\left(N\right)=T_{\text{biggest}}$$

## 5 Evaluation

In this section, we discuss our scheme's communication cost and training effect and provide comparisons with Paillier scheme from different dimensions under the same security key length of 4096. By the way, the Paillier library we used comes from [8] and the CKKS library we use is TenSEAL [2].

### 5.1 Experimental Setup

The server's CPU is Intel(R) Xeon(R) Silver 4210R @ 2.40GHz and GPU is Tesla V100S and the IoT devices we used both are Jetson AGX Xavier Developer Kit. We simulate the scenario of one server and two participants and use logistic regression to train a model on MINIST dataset. The model parameters constain a weight matrix of $10 \times 784$ and a bias vector of $1 \times 10$.

### 5.2 Communication Cost

Since ciphertext accounts for a large part of the communication cost, we discuss the memory cost of ciphertext first and then we verify our communication cost analysis provided in Sect. 4.3 according to experimental results. It is worth mentioning that all serialized ciphertext data in this scheme are converted into strings through Base64 encoding and transmitted in JSON format.

**Memory Cost of Ciphertext.** As shown in Fig. 3, the CKKS's memory space of ciphertext increases in stages and the Paillier's memory space increases linearly as the number of parameters grows. We can also find that the Paillier [18] scheme are obviously worse than the CKKS scheme in memory space when the number of parameters is big enough. Table 1 shows the comparison before and after CKKS encryption in different number of parameters and gives the corresponding expansion multiple of list after encryption, from which we can see that CKKS's memory space increases about 126 KB every $2 \times 10^3$ parameters added and the memory space is approximately about 126 KB if the number of parameters is smaller than $2 \times 10^3$.

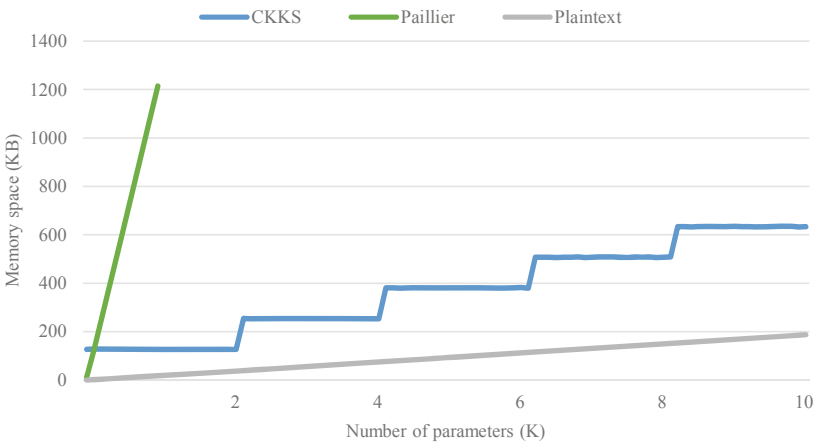



**Fig. 3.** Comparison in memory among serialized ciphertext from CKKS, Paillier and plaintext.

**Experimental Result.** Table 2 shows the total communication cost before and after encryption from which we can get an approximate expansion multiple of the communication cost, 8.5. We can verify our mathematical theory according Table 1 and the process can be presented as follows:

$$\frac{7840}{7850} \times 8.602775 + \frac{10}{7850} \times 667.953608 \approx 9.442712$$

Considering other communication cost $l$, the experimental result is basically consistent with the theory.

**Table 1.** Comparison and analysis of memory space before and after CKKS encryption with different number of parameters.

| Number | Before Enc. (Byte) | After Enc. (Byte)/(KB) | Multiple |
|---|---|---|---|
| 2000 | 38356 | 129490/126.4 | 3.376003 |
| 4000 | 76704 | 259328/253.2 | 3.380892 |
| 6000 | 115017 | 390743/381.5 | 3.397263 |
| 8000 | 153468 | 519305/507.1 | 3.383799 |
| 784 | 15062 | 129575/126.5 | 8.602775 |
| 10 | 194 | 129583/126.5 | 667.953608 |
| 7850 | 150530 | 519252/507.8 | 3.449491 |

**Table 2.** Comparison and analysis of network traffic before and after CKKS encryption.

| Data | Before Enc. (Byte) | After Enc. (Byte) | Multiple |
|---|---|---|---|
| Send | 491041876 | 57839950 | 8.489666 |
| Receive | 490643031 | 57544559 | 8.526315 |

### 5.3 Evaluatioin on the Training Performance

We train model for 80 rounds and record the training results before and after encryption. The records are shown in Table 3. There is no much difference between PE-FedAvg scheme and FedAvg scheme except the training time. The time cost increases by nearly 20% which is acceptable for a server. Compared with Paillier scheme, CKKS' time cost in computation is far more smaller as shown in Table 4. We can find that the ciphertext addition time and encryption and decryption time of CKKS scheme are three orders of magnitude smaller than

**Table 3.** Comparison of training effect between PE-FedAvg scheme and FedAvg scheme.

| Model | FedAvg | PE-FedAvg |
|---|---|---|
| Train Accuracy | 82.52% | 81.98% |
| Train Loss | 1.7349 | 1.6959 |
| Test Accuracy | 83.06% | 82.07% |
| Test Loss | 1.7363 | 1.6918 |
| Aggregation Time | 207s | 243s |

**Table 4.** Comparison of computation cost between CKKS scheme and Paillier scheme.

| Scheme | Addition | Encryption | Decryption |
|---|---|---|---|
| CKKS | 0.031s | 0.006s | 0.002s |
| Paillier | 11.924s | 58.200s | 16.720s |

that of Paillier scheme. After decryption, these parameters are approximate to original data, so here is a little difference in accuracy which is ignorable.

In summary, our scheme PE-FedAvg enhances the privacy of federated learning and gets an ideal training results as FedAvg scheme do. Besides, our scheme has a nice performance on computation efficiency compared with Fang and Qian's Paillier scheme [10].

## 6 Conclusion and Future Work

In this paper, we propose a scheme called PE-FedAvg which based on FedAvg and CKKS. The scheme can protect participants' model parameters away from gradients attack and also defense the quantum attack in a way. We then give detailed design and algorithm of our scheme. After that, we simulate one server and two participants scenario on a machine and analyze the communication cost and training effect before and after encryption. Finally, we discuss the feasibility of deploying it on IoT scenario and the experimental results show that this scheme PE-FedAvg is suitable for it.

However, the scheme's security assumption is that all the participants are honest and the server is honest but curious which means it can't defense the collusion attack between the server and one vicious device. We can solve this problem by dividing secret key into many parts and distributing them to participants as Ma et al. [15] do. Besides, we can also get a lower communication cost by making an engineering improvement theoretically. We can transform weights $N \times M$ and bias $1 \times N$ to a list sized of $NM + N$ from which we can get a lower the expansion multiple. From Table 1, we can get approximate 3.44 multiple on a list sized of 7850 which is better than 9.44 multiple in our scheme. Therefore, our further work is to improve the security of defensing collusion attack and

make an engineering improvement on parameters to reduce the communication cost.

**Acknowledgment.** This work was supported by the National Key R&D Program of China (2020AAA0107703), the National Natural Science Foundation of China (62132008, 62071222, U20A201092, U20A20176) and the Natural Science Foundation of Jiangsu Province (Grant No.BK20200418).

## References

1. Aono, Y., Hayashi, T., Wang, L., Moriai, S., et al.: Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans. Inf. Forensic. Secur. **13**(5), 1333–1345 (2017)
2. Benaissa, A., Retiat, B., Cebere, B., Belfedhal, A.E.: Tenseal: a library for encrypted tensor operations using homomorphic encryption. arXiv preprint arXiv:2104.03152 (2021)
3. Bonawitz, K., et al.: Practical secure aggregation for federated learning on user-held data. arXiv preprint arXiv:1611.04482 (2016)
4. Bonawitz, K., et al.: Practical secure aggregation for privacy-preserving machine learning. In: proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1175–1191 (2017)
5. Chen, H., Chillotti, I., Song, Y.: Improved bootstrapping for approximate homomorphic encryption. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11477, pp. 34–54. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_2
6. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: Bootstrapping for approximate homomorphic encryption. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 360–384. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_14
7. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 409–437. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_15
8. Data61, C.: Python paillier library (2013). https://github.com/data61/python-paillier
9. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79228-4_1
10. Fang, H., Qian, Q.: Privacy preserving machine learning with homomorphic encryption and federated learning. Future Internet **13**(4), 94 (2021)
11. He, C., et al.: FedML: a research library and benchmark for federated machine learning. arXiv preprint arXiv:2007.13518 (2020)
12. Hitaj, B., Ateniese, G., Perez-Cruz, F.: Deep models under the GAN: information leakage from collaborative deep learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, pp. 603–618 (2017)
13. Konečnỳ, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)

14. Lin, Y., Han, S., Mao, H., Wang, Y., Dally, W.J.: Deep gradient compression: reducing the communication bandwidth for distributed training. arXiv preprint arXiv:1712.01887 (2017)
15. Ma, J., Naas, S.A., Sigg, S., Lyu, X.: Privacy-preserving federated learning based on multi-key homomorphic encryption. Int. J. Intell. Syst. (2022)
16. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics, pp. 1273–1282. PMLR (2017)
17. Melis, L., Song, C., De Cristofaro, E., Shmatikov, V.: Exploiting unintended feature leakage in collaborative learning. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 691–706. IEEE (2019)
18. Paillier, P.: Public-Key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
19. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 3–18. IEEE (2017)
20. Statista. https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/. Accessed 27 Nov 2016
21. Tsuzuku, Y., Imachi, H., Akiba, T.: Variance-based gradient compression for efficient distributed deep learning. arXiv preprint arXiv:1802.06058 (2018)
22. Zhao, J., Chen, Y., Zhang, W.: Differential privacy preservation in deep learning: challenges, opportunities and solutions. IEEE Access **7**, 48901–48911 (2019)
23. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. Adv. Neural Inf. Proc. Syst. 32 (2019)

# Reinforcement Learning Based Vulnerability Analysis for Smart Grids Against False Data Injection Attacks

Shiyu Xu, Shi Yu, Liang Xiao(✉), and Zefang Lv

Department of Information and Communication Engineering, Xiamen University, Xiamen, China
lxiao@xmu.edu.cn

**Abstract.** False data injection attacks modify the meter measurements to mislead the control center into estimating inaccurate system states and thus affect the reliable operation of smart grids. In this paper, we propose a deep reinforcement learning based vulnerability analysis scheme for smart grids that enables the control center to construct an attack vector from the attacker's view to identify the vulnerable meters. The control center chooses the attack vector based on power system states, meter measurements, the previous number of analyzed meters, and injected errors without knowing the power system topology. This scheme designs an actor-critic architecture that applies an actor network to output the policy probability distribution to handle the continuous and high-dimensional vulnerability analysis policy and contains a critic network to guide the weights update of the actor network. We also analyze the computational complexity and perform simulations to verify the efficacy of this scheme in terms of the vulnerability detection rate, the number of analyzed meters and the utility.

**Keywords:** Smart grids · Vulnerability analysis · Reinforcement learning · False data injection attacks

## 1 Introduction

Smart meters placed in buses or transmission lines measure the power flows to support smart grid applications such as automatic voltage control and energy management. However, the transmission of meter measurement has to address false data injection attacks that inject malicious measurement to each actual measurement based on the vulnerability of smart grids. The attack vector containing the injected data is chosen to mislead power system states such as the bus voltage angles in state estimation, cause line outages to affect the stability of the power grid [1], and obtain illegal financial profits by manipulating the electricity market price [2].

Smart grids have to identify the vulnerable meters that tend to be injected altered data but will not be detected under false data injection attacks. The


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 441–452, 2022.
https://doi.org/10.1007/978-3-031-19208-1_36

control center of the smart grid can construct a vulnerability analyzer from the attacker's view to investigate the vulnerable meters. With the aim of maximizing the vulnerability detection rate while minimizing the number of analyzed meters, the analyzer constructs the attack vector to simulate the attack process in state estimation and perform the vulnerability analysis. Then the control center applies the traditional bad data detection techniques such as the residual test [3] to evaluate whether the simulated attack succeeds.

The attack vector determines the injected errors and the vulnerability detection rate of the vulnerability analyzer in the control center. Reinforcement learning (RL) is applied to construct the attack vector without knowing the power system topology. In particular, a Q-learning based vulnerability analysis scheme LA as proposed in [4] chooses the injected data on the actual meter measurements based on the bus voltage angles and amplitudes, active and reactive power of the generator and the loads to identify the vulnerable meters in smart grids, but suffers from high detection delay under large-scale grids. A double deep Q-network based vulnerability analysis scheme DLA as proposed in [5] applies prioritized experience replay to label the priority of vulnerability analysis experiences in the replay pool to accelerate the detection speed and improve the learning effectiveness in large-scale grids. However, DLA has vulnerability detection rate degradation if the power flow continuously changes.

In this paper, we propose a vulnerability analysis scheme based on deep RL to explore the vulnerabilities of state estimation and improve the security of smart grids with numerous meters. The proposed scheme designs an actor-critic architecture with lightweight fully connected (FC) layers to handle the continuous and high-dimensional vulnerability analysis policy. The actor-critic architecture contains the actor network that chooses the vulnerability analysis policy, i.e., the attack vector, based on bus voltage angles, meter measurements, the previous number of analyzed meters and the previous injected errors and the critic network that outputs the value of the observed state to update the actor network weights towards the direction of analysis performance improvement.

The computational complexity of this scheme is analyzed and the simulations with the IEEE 14 bus system that consists of 54 meters and 13 system states are performed to show that the vulnerability detection rate and the utility are improved with fewer number of the analyzed meters compared with DLA [5] and LA [4].

The contributions of this paper are summarized as follows:

1) We propose a vulnerability analysis scheme to optimize the attack vector based on power system states, meter measurements, the previous number of analyzed meters, and injected errors to explore the vulnerabilities of state estimation without knowing the power system topology.

2) We present a deep RL based vulnerability analysis algorithm that designs an actor-critic architecture to compress high-dimensional states to further improve the analysis performance with continuously changing power flows and system states over time.

3) The computational complexity of the designed algorithm is analyzed and simulation results with an IEEE 14 bus system prove that the proposed scheme can identify vulnerabilities of smart grids with a higher vulnerability detection rate and less number of analyzed meters.

## 2 Related Work

With the full acknowledgment of the grid topology and parameter information such as line admittance, the attacker constructs the attack vector to implement the stealthy data injection attacks [6–8]. For example, a convex relaxation based attack scheme as proposed in [6] chose the optimal attack vector to maximize the injected errors and avoid being detected based on the topology of smart grids. A convex optimization based sparse attack vector was constructed to decrease the probability of being detected and the number of compromised meters based on the Jacobian matrix for random attackers and targeted attackers [7]. The limited resource attacker applied interior point methods and the Jacobian matrix to find the sparsest attack vector and evade bad data detection successfully when a subset of meters was protected [8].

False data injection attackers also use incomplete grid system information to compromise the measurements. For instance, the attacker used scenario generation to select the attack vector and reduce the probability of being detected while maximizing the injected errors based on the probability distribution of the transmission line admittance values [9]. A local load redistribution attack scheme as proposed in [10] chose the attack vector based on the attacking region information to avoid being detected. An attack scheme as proposed in [11] constructed the attack vector to modify the bus or superbus state variables based on limited transmission-line susceptance information.

RL has been used to improve the vulnerability detection performance. For example, the Q-learning based vulnerability analysis scheme as proposed in [4] constructed the attack vector with the aim of reducing the bus voltages to provide the power system security assessment and improvement. The vulnerability analysis scheme against sequential topology attacks in [12] applied Q-learning to choose the attacking line to explore more vulnerable lines with less attack cost. The deep RL based vulnerability analysis scheme in [5] chose the injected data to increase the probability of a successful attack.

## 3 System Model

### 3.1 Network Model

In the smart grid as shown in Fig. 1, $M$ smart meters send their measurements to support the automatic voltage control and energy management. The control center performs the vulnerability analysis to investigate the vulnerable meters and guarantee accurate power flow measurements on buses and transmission lines against false data injection attacks.

The control center uses smart meter measurements such as power flow to monitor the real-time operation of smart grids. At time slot $k$ lasting $T$ seconds, smart meter $i \in \{1, 2, \cdots, M\}$ sends $Z$-bit measurements $z_i^{(k)}$ to the control center to estimate $N$ power system states, i.e., $N$ bus voltage angles.

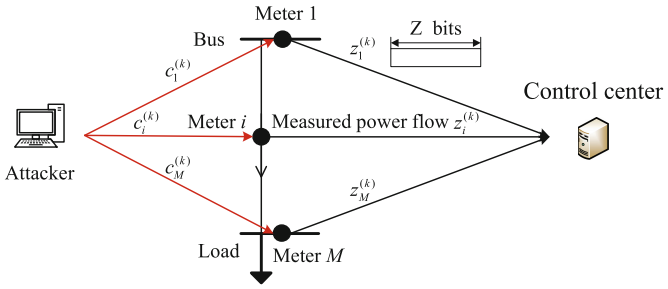



**Fig. 1.** Illustration of a smart grid, in which smart meters transmit meter measurements $[z_i^{(k)}]_{1 \le i \le M}$, the control center performs the vulnerability analysis to secure the vulnerable meters and the attacker compromises the meter measurements with $[c_i^{(k)}]_{1 \le i \le M}$ to mislead the control center

The state estimator in the control center formulates a $M \times N$ network topology matrix denoted by $\boldsymbol{H} = [h_{i,j}]_{1 \le i \le M, 1 \le j \le N}$ according to the direction of power flows and the admittance of transmission lines, with each element $h_{i,j}$ representing the line impedances of the grid system. According to the typical direct current power flow model and the maximum likelihood based method in [13], the state estimator evaluates the bus voltages angles $\boldsymbol{x}^{(k)} = [x_j^{(k)}]_{1 \le j \le N}$ based on the network topology matrix $\boldsymbol{H}$ and the meter measurements $\boldsymbol{z}^{(k)} = [z_i^{(k)}]_{1 \le i \le M}$, given by

$$\boldsymbol{x}^{(k)} = \left(\boldsymbol{H}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{H}\right)^{-1} \boldsymbol{H}^{\mathrm{T}} \boldsymbol{W} \boldsymbol{z}^{(k)}, \tag{1}$$

where $\boldsymbol{W}$ is a $M \times M$ diagonal matrix representing reciprocals of the variances of meter errors.

Based on the measurements $\boldsymbol{z}^{(k)}$, bus voltages angles $\boldsymbol{x}^{(k)}$, and the detection result in the last time slot, the vulnerability analyzer in the control center evaluates the number of analyzed meters and the injected errors, and constructs the attack vector $\boldsymbol{a}^{(k)} = [a_i^{(k)}]_{1 \le i \le M}$ to explore the potential risks in smart grids and the corresponding countermeasures. By adding the attack vector $\boldsymbol{a}^{(k)}$ to $M$ meter measurements, the altered measurements $\overline{\boldsymbol{z}}^{(k)} = \boldsymbol{z}^{(k)} + \boldsymbol{a}^{(k)}$ are used to estimate the $N$ bus voltage angles $\overline{\boldsymbol{x}}^{(k)}$ similar to (1) and perform the vulnerability detection.

According to the bus voltage angles $\overline{\boldsymbol{x}}^{(k)}$ and the altered measurement $\overline{\boldsymbol{z}}^{(k)}$, the control center applies the bad data detection technique, such as the measurement residual test, to identify whether the injected data can be found. More

specifically, the control center calculates the $l$-2 norm difference $\Delta$ between the measurements $\overline{\boldsymbol{z}}^{(k)}$ and the estimated measurements $\boldsymbol{H}\overline{\boldsymbol{x}}^{(k)}$ and compares $\Delta$ with the detection threshold $\omega$ determined by Q-learning. If $\Delta \leq \omega$, the meters with nonzero injection data in the attack vector are potential to be attacked and should be protected. Otherwise, the control center successfully detects the simulated attacks launched by the vulnerability analyzer and the injected data are eliminated.

Upon the detection is finished, the control center sends the detection result 0 or 1 to the analyzer, where 0 represents that the simulated attack of the analyzer is not detected by the control center and 1 otherwise. If the detection result is 0, the control center will take protective measures, such as encryption and continuous monitoring, against the discovered vulnerable meters.

### 3.2 Attack Model

To distort the state estimation, the false data injection attacker exhausts smart meters' bandwidth to disrupt the connection between smart meters and the control center and then compromises the meter measurements in physical memory by injecting malicious code into smart meters [14]. Such attacker injects compromised measurements to stealthily bias the power system states, further resulting in physical damages such as blackout, and obtaining illegal benefits.

The attacker is assumed to randomly modify $y$ meter measurements to mislead the estimation of the bus voltages angles with the limited resources, i.e., $y \leq Y$, where $Y$ is the maximum number of compromised meter measurements. The attacker formulates the attack vector $\boldsymbol{c}^{(k)} = [c_i^{(k)}]_{1 \leq i \leq M} \in [0, D]^M$ with maximum injected data $D$ and injects $c_i^{(k)}$ to the meter $i$ to alter $M$ meter measurements $\boldsymbol{z}^{(k)}$. Considering the limited resources of the attacker, there are at most $Y$ nonzero elements in the attack vector. The superscript $k$ is omitted if no confusion incurs.

## 4 Deep RL Based Vulnerability Analysis

A deep RL based vulnerability analysis scheme (DRVA) is proposed to improve the vulnerability detection rate and the response speed for the smart grids with a large number of smart meters. This scheme designs the actor-critic architecture consisting of an actor network and a critic network for the control center to handle the continuous and high-dimensional vulnerability analysis policies and thus reduce the quantization errors of the injected data. In particular, the actor network chooses the attack vector $\boldsymbol{a}^{(k)}$ based on the estimated bus voltage angles, the received meter measurements, the previous number of analyzed meters, and the previous injected errors. The critic network evaluates the state value to update the actor network weights. The actor-critic structure uses the lightweight FC layers instead of the convolutional layers to extract the analysis features and thus accelerate the optimization speed.

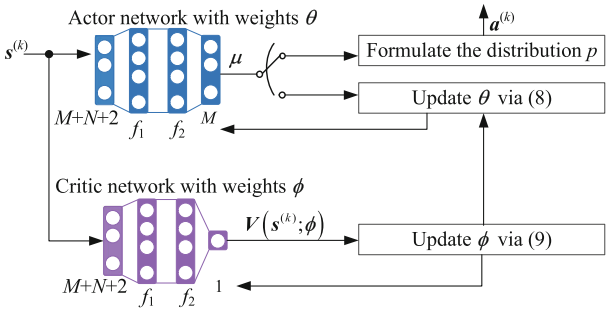



**Fig. 2.** Illustration of the proposed DRVA algorithm for the control center in smart grids

**Algorithm 1** Actor-critic based vulnerability analysis scheme

**Initialize:** $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, $\gamma$, $\tau$, $\boldsymbol{\sigma}^2$, $r$ and $f$
**for** $k = 1, 2, \ldots$ **do**
  Receive $[z_i]_{1 \le i \le M}$
  Estimate $[x_j]_{1 \le j \le N}$
  Formulate the state $\boldsymbol{s}^{(k)}$ via (2)
  Input $\boldsymbol{s}^{(k)}$ to the actor network and output $\boldsymbol{\mu}$
  Generate the Gaussian probability distribution $p(\cdot|\boldsymbol{s}^{(k)};\boldsymbol{\mu},\boldsymbol{\sigma}^2)$ based on $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$
  Choose the attack vector $\boldsymbol{a}^{(k)} = [a_i^{(k)}]_{1 \le i \le M}$ according to the distribution $p(\cdot|\boldsymbol{s}^{(k)};\boldsymbol{\mu},\boldsymbol{\sigma}^2)$
  Input $\boldsymbol{s}^{(k)}$ to the critic network and output $V\left(\boldsymbol{s}^{(k)};\boldsymbol{\phi}\right)$
  Calculate $f$
  Evaluate $r$ and $\tau$
  Obtain $u^{(k)}$ via (4)
  Obtain the discount-accumulated reward $R$ via (5)
  Calculate the advantage and loss functions via (6) and (7)
  Update the network weights $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ with Adam via (8) and (9)
**end for**

Upon receiving the measurements $[z_i]_{1 \le i \le M}$ transmitted from smart meters to the control center, the vulnerability analyzer applies the maximum-likelihood estimation [15] to estimate the bus voltage angles $[x_j]_{1 \le j \le N}$ and formulates the state $\boldsymbol{s}^{(k)}$ consisting of the above information, the number of analyzed meters $r$ and injected errors $\tau$ in the previous time slot, i.e.,

$$\boldsymbol{s}^{(k)} = \left[[z_i]_{1 \le i \le M}, [x_j]_{1 \le j \le N}, r, \tau\right]. \tag{2}$$

The state $\boldsymbol{s}^{(k)}$ is input to the actor network with weights $\boldsymbol{\theta}$ that consists of four FC layers, each with $M + N + 2$, $f_1$, $f_2$, $M$ units as shown in Fig. 2. Then the actor network outputs the $M$-dimensional mean value of the analysis policy $\boldsymbol{\mu} = [\mu_i]_{1 \le i \le M}$. According to $\boldsymbol{\mu}$ and the given $M$-dimensional variance

of the analysis policy $\boldsymbol{\sigma}^2 = [\sigma_i^2]_{1 \le i \le M}$, a Gaussian probability distribution $p\big(\cdot \mid \boldsymbol{s}^{(k)}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\big)$ is given by

$$p\left(\cdot \mid \boldsymbol{s}^{(k)}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\right) = \frac{\exp\left(-\sum\limits_{i=1}^{M} \frac{(a_i - \mu_i)^2}{2\sigma_i^2}\right)}{(2\pi)^{M/2} \prod\limits_{i=1}^{M} \sigma_i^2}. \tag{3}$$

Then the vulnerability analyzer chooses the $M$-dimension attack vector $\boldsymbol{a}^{(k)} = [a_i]_{1 \le i \le M}$ based on the distribution $p\big(\cdot \mid \boldsymbol{s}^{(k)}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\big)$ and injects $a_i$ to the meter measurement $z_i$ to identify the vulnerabilities of smart grids.

The critic network consists of four FC layers with $M + N + 2$, $f_1$, $f_2$, 1 units, whose input is the state $\boldsymbol{s}^{(k)}$. The output layer of the critic network evaluates the value of the input state $V\left(\boldsymbol{s}^{(k)}; \boldsymbol{\phi}\right)$, where $\boldsymbol{\phi}$ is the critic network weights. The activation function of the FC layers in the actor and critic network is rectified linear unit.

The analyzer calculates the nonzero elements in $\boldsymbol{a}^{(k)}$ as the number of analyzed meters $r$ and estimates the injected errors of state estimation $\tau$ based on the $l$-2 norm multiplication of the topology matrix and the attack vector using the maximum likelihood based method in [13]. Upon receiving the detection results from the control center, the vulnerability detection rate $f$ is calculated based on the vulnerability detection results in the previous $m$ time slots. The vulnerability analyzer expects to increase the injected errors $\tau$, the vulnerability detection rate $f$, and reduce the number of analyzed meters $r$. Thus the utility $u^{(k)}$ is modeled by

$$u^{(k)} = \tau + c_1 f - c_2 r, \tag{4}$$

where coefficients $c_1$ and $c_2$ represent the importance of the vulnerability detection rate and the number of analyzed meters, respectively.

The discounted accumulated reward $R$ is accumulated by

$$R = u^{(k)} + \gamma V\left(\boldsymbol{s}^{(k)}; \boldsymbol{\phi}\right), \tag{5}$$

where $\gamma \in [0, 1]$ represents the discount factor. The vulnerability analyzer maximizes the discounted accumulated reward to increase the vulnerability detection rate and the utility, and reduce the number of analyzed meters. The advantage function used to decrease the variance of reward $R$ at state $\boldsymbol{s}^{(k)}$ is defined as

$$A\left(\boldsymbol{s}^{(k)}; \boldsymbol{\phi}\right) = R - V\left(\boldsymbol{s}^{(k)}; \boldsymbol{\phi}\right). \tag{6}$$

The loss functions of the actor network is given by

$$\begin{aligned} L\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}; \boldsymbol{\theta}\right) = &- \log\left(p\Big(\cdot \mid \boldsymbol{s}^{(k)}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\Big)\right) A\left(\boldsymbol{s}^{(k)}; \boldsymbol{\phi}\right) \\ &- \beta H\left(p\Big(\cdot \mid \boldsymbol{s}^{(k)}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\Big)\right), \end{aligned} \tag{7}$$

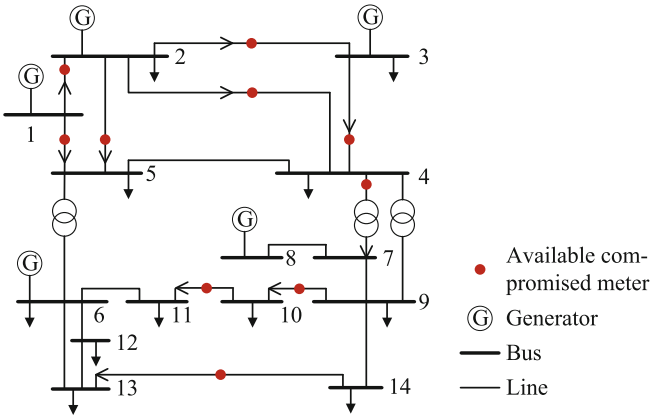



**Fig. 3.** IEEE 14 bus system topology

where $H\left(p\left(\cdot \mid \boldsymbol{s}^{(k)}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\right)\right)$ represents the entropy of the analysis policy distribution $p\left(\cdot \mid \boldsymbol{s}^{(k)}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\right)$ to encourage the policy exploration, and $\beta \in [0, 1]$ is the factor controlling the strength of entropy regularization term.

The policy gradients method, i.e., Adam optimizer [16], is used to update the actor network weights $\boldsymbol{\theta}$ and the critic network weights $\boldsymbol{\phi}$ as shown in Algorithm 1. According to the Adam optimizer and the loss functions in (7) , the actor network weights are updated by minimizing the loss function $L\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}; \boldsymbol{\theta}\right)$, i.e.,

$$\boldsymbol{\theta} = \arg\min_{\boldsymbol{\theta}'} \mathbb{E}\left[L\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}; \boldsymbol{\theta}'\right)\right], \tag{8}$$

where the $\mathbb{E}[\cdot]$ is the expected value function. The critic network weights are updated by minimizing the advantage function, i.e.,

$$\boldsymbol{\phi} = \arg\min_{\boldsymbol{\phi}'} \mathbb{E}\left[A^2\left(\boldsymbol{s}^{(k)}; \boldsymbol{\phi}'\right)\right]. \tag{9}$$

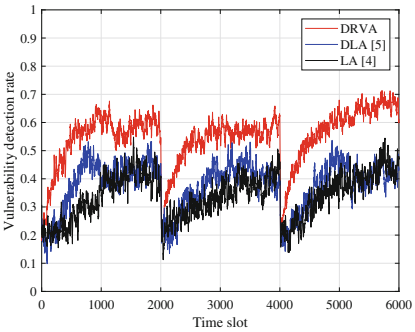
## 5 Complexity Analysis

According to [17], the computational complexity of **Algorithm** 1 consists of the forward and backward calculations of the number of multiplications for two FC layers. The forward multiplications of the actor network is calculated as

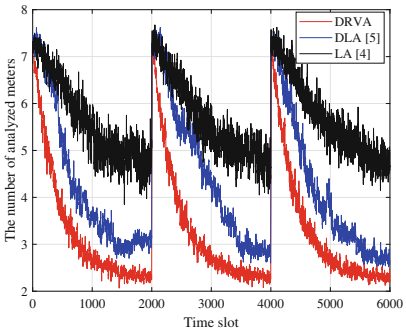
$$W_{\mathrm{a}} = f_1\left(M + N + 3\right) + f_2\left(f_1 + 1\right) + M\left(f_2 + 1\right). \tag{10}$$

The calculation of the backward multiplications in the actor network is given by

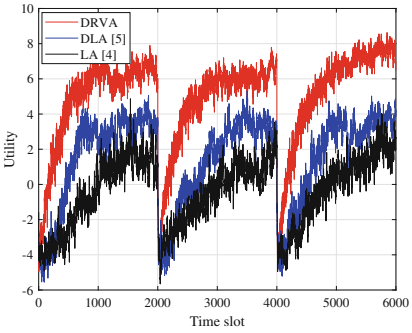
$$W_{\mathrm{b}} = 2f_1\left(M + N + 3\right) + 2f_2\left(f_1 + 1\right) + 3M\left(f_2 + 1\right). \tag{11}$$




(a) Vulnerability detection rate




(b) The number of analyzed meters




(c) Utility

**Fig. 4.** Performance of the proposed scheme, in which the detection threshold of the control center is chosen from 0.3 to 1.1, the analyze area of smart meters changes every 2000 time slots in smart grids, and the data injection attacker randomly compromises 2 meters

Similarly, the forward and the backward calculations in the critic network are given by

$$W_{\mathrm{c}} = f_1\left(M + N + 3\right) + f_2(f_1 + 1) + f_2 + 1, \tag{12}$$

$$W_{\mathrm{d}} = 2f_1\left(M+N+3\right) + 2f_2(f_1+1) + 3f_2 + 3. \tag{13}$$

According to [18], the number of units required to learn $K$ samples in the first and second FC layer are given by

$$f_1 = \sqrt{KM} + 2\sqrt{\frac{K}{M}} \tag{14}$$

and

$$f_2 = \sqrt{KM}. \tag{15}$$

Hence, by (10–15), the computational complexity of the **Algorithm** 1 is

$$\begin{aligned} W &= \mathcal{O}\big(W_{\mathrm{a}} + W_{\mathrm{b}} + W_{\mathrm{c}} + W_{\mathrm{d}}\big) \\ &= \mathcal{O}\big(6f_1\left(M+N+3\right) + 6f_2\left(f_1+1\right) + 4M\left(f_2+1\right) \\ &\quad + 4f_2 + 4\big) \\ &= \mathcal{O}\big(6f_1\left(M+N\right) + 6f_1f_2 + 4Mf_2\big) \\ &= \mathcal{O}\big(\left(10M+6N\right)\sqrt{KM} + 6KM\big) \\ &= \mathcal{O}\big(KM\big). \end{aligned} \tag{16}$$

## 6 Simulation Results

Simulations were performed to validate the efficacy of the DRVA with an IEEE 14 bus system. As shown in Fig. 3, IEEE 14 bus system includes 14 buses and 20 transmission lines, and 54 meters are placed in buses and transmission lines, 14 of which are utilized to collect power flows of buses and 40 of which are used to measure the power flows of transmission lines. The analyzed meter region changes every 2000 time slots and the power flows of 54 meters change every time slot with the Gaussian measurement noise $N\left(0, 0.05\right)$. The power system states consisting of 13 bus voltage angles are used to monitor the operation state of the grid. The vulnerability analyzer of the control center injects power flows ranging from 0.3 to 0.5 MW and the number of analyzed meters is chosen from $\{2, 4, 6, 8\}$. The vulnerability detection rate is calculated based on the vulnerability detection results in the previous 10 time slots. The control center applies Q-learning to choose the detection threshold ranging from 0.3 to 1.1. The data injection attacker randomly compromises 2 meters in the analyzed meter region. The number of units in the actor and critic networks $f_1 = f_2 = v_1 = v_2 = 128$ and the discount factor $\gamma = 0.5$ and the learning rate $\alpha = 0.001$.

As shown in Fig. 4, the DRVA improves the vulnerability detection performance including the detection rate and the number of analyzed meters. For instance, the vulnerability detection rate of DRVA increases 162.7% and the number of analyzed meters decreases 68.7% at 4000-th time slot. The actor-critic architecture in DRVA mitigates the policy quantization error and thus improves the vulnerability detection performance compared with DLA and LA with higher vulnerability detection rate and fewer analyzed meters. For example, the vulnerability detection rate is 33.1% and 40.3% higher than DLA and

LA, respectively. Meantime, the number of analyzed meters decreases 19.4% and 51.3% compared with DLA and LA. As a result, the utility of the analyzer outperforms DLA and LA 76.5% and 170% at 4000-th time slot, respectively.

## 7 Conclusion

In this paper, we have proposed a deep RL based vulnerability analysis scheme for the control center of the smart grid to resist false data injection attacks. This scheme designs an actor-critic architecture to optimize the attack vector from the attacker's view and thus identify the vulnerable meters without requiring the topology information of the grid system. The computational complexity of this scheme is analyzed and simulation results of an IEEE 14 bus system show that our proposed scheme improves the vulnerability detection performance and outperforms the benchmark schemes. For example, the proposed analysis scheme improves 33.1% vulnerability detection rate and 76.5% utility, and reduces 19.4% number of analyzed meters after 2000 time slots compared with DLA in [5].

**Acknowledgements.** This work was supported in part by the Natural Science Foundation of China under Grant 61971366 and Grant U21A20444.

## References

1. Liu, X., Li, Z., Liu, X., Li, Z.: Masking transmission line outages via false data injection attacks. IEEE Trans. Inf. Forensics Security **11**(7), 1592–1602 (2016)
2. Zhang, Q., Li, F., Shi, Q., Tomsovic, K., Sun, J., Ren, L.: Profit-oriented false data injection on electricity market: reviews, analyses, and insights. IEEE Trans. Ind. Informat. **17**(9), 5876–5886 (2020)
3. Gu, C., Panida, J., Mehul, M.: Detecting false data injection attacks in ac state estimation. IEEE Trans. Smart Grid **6**(5), 2476–2483 (2015)
4. Chen, Y., Huang, S., Liu, F., Wang, Z., Sun, X.: Evaluation of reinforcement learning-based false data injection attack to automatic voltage control. IEEE Trans. Smart Grid **10**(2), 2158–2169 (2018)
5. Luo, W., Xiao, L.: Reinforcement learning based vulnerability analysis of data injection attack for smart grids. In: 2021 40th Chinese Control Conference (CCC), pp. 6788–6792. IEEE (2021). http://orcid.org/10.23919/CCC52363.2021.9550523
6. Xie, L., Mo, Y., Sinopoli, B.: Integrity data attacks in power market operations. IEEE Trans. Smart Grid **2**(4), 659–666 (2011)
7. Hao, J., Piechocki, R.J., Kaleshi, D., Chin, W.H., Fan, Z.: Sparse malicious false data injection attacks and defense mechanisms in smart grids. IEEE Trans. Ind. Informat. **11**(5), 1–12 (2015)
8. Kim, T.T., Poor, H.V.: Strategic protection against data injection attacks on power grids. IEEE Trans. Smart Grid **2**(2), 326–333 (2011)
9. Rahman, M.A., Mohsenian-Rad, H.: False data injection attacks with incomplete information against smart power grids. In: 2012 IEEE Global Communications Conference (GLOBECOM), pp. 3153–3158 (2012). http://orcid.org/10.1109/GLOCOM.2012.6503599

10. Liu, X., Bao, Z., Lu, D., Li, Z.: Modeling of local false data injection attacks with reduced network information. IEEE Trans. Smart Grid **6**(4), 1686–1696 (2015)
11. Deng, R., Liang, H.: False data injection attacks with limited susceptance information and new countermeasures in smart grid. IEEE Trans. Ind. Informat. **15**(3), 1619–1628 (2018)
12. Yan, J., He, H., Zhong, X., Tang, Y.: Q-learning-based vulnerability analysis of smart grid against sequential topology attacks. IEEE Trans. Inf. Forensics Security **12**(1), 200–210 (2016)
13. Yu, Z.H., Chin, W.L.: Blind false data injection attack using PCA approximation method in smart grid. IEEE Trans. Smart Grid **6**(3), 1219–1226 (2015)
14. Liu, X., Zhu, P., Zhang, Y., Chen, K.: A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure. IEEE Trans. Smart Grid **6**(5), 2435–2443 (2015)
15. Liang, G., Zhao, J., Luo, F., Weller, S.R., Dong, Z.Y.: A review of false data injection attacks against modern power systems. IEEE Trans. Smart Grid **8**(4), 1630–1638 (2016)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 2015, Conference Track Proceedings (2015). arxiv.org/abs/1412.6980
17. Ou, G., Murphey, Y.L.: Multi-class pattern classification using neural networks. Pattern Recogn. **40**(1), 4–18 (2007)
18. Huang, G.B.: Learning capability and storage capacity of two-hidden-layer feedforward networks. IEEE Trans. Neural Netw. **14**(2), 274–281 (2003)

# Blockchain-Based Secure and Efficient Federated Learning with Three-phase Consensus and Unknown Device Selection

Jianrong Wang[1,3], Haoran Sun[2,3], and Tianyi Xu[1,3(✉)]

[1] College of Intelligence and Computing, Tianjin University, Tianjin, China
wjr@tju.edu.cn, tianyi.xu@tju.edu.cn
[2] Tianjin International Engineering Institute, Tianjin University, Tianjin, China
shy5500501@tju.edu.cn
[3] Tianjin Key Laboratory of Advanced Networking (TANK Lab), Tianjin, China

**Abstract.** Blockchain-based decentralized federated learning (BCFL) protects data privacy and avoids the single point of failure, which has become a key technology in the Intelligent Internet of Things application. However, the BCFL is still challenged by model attacks from malicious devices. In addition, although the BCFL has many schemes for selecting candidate devices, most of them either assume that devices qualities are known in advance or cannot ensure devices to report information honestly. This will lead to inefficient training in the face of unknown candidate device. To address these issues, we consider two types of model attacks and design "Proof of Model Quality" (PoMQ) to protect the security of BCFL. The PoMQ is a three-phase consensus algorithm, which combines the FedAvg and Multi-Krum algorithms to defend against model update attacks. The PoMQ is also based on the model verification process to resist model aggregation attacks. Moreover, we define the unknown device selection problem as a Combinatorial Multi-Armed Bandit problem (CMAB) to improve the training efficiency, and propose an online learning algorithm based on PoMQ (OLAC) to solve CMAB. Finally, through analysis and extensive experiments, we prove that PoMQ and OLAC can better improve the robustness and training efficiency of BCFL.

**Keywords:** Blockchain · Federated learning · Consensus algorithm · Multi-armed bandit

This study was supported by the National Natural Science Foundation Council of China under Project 92167206 and the Open Research Project of Zhejiang Lab (No. 2021KF0AB02).


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 453–465, 2022.
https://doi.org/10.1007/978-3-031-19208-1_37

## 1 Introduction

With the oncoming of Internet of Things, massive data generated from various connected devices (e.g., mobile phones, vehicles, and smart sensors) has been regarded as a valuable treasure to serve future society [1]. Meanwhile, Blockchain-based decentralized federated learning (BCFL) as a promising training paradigm has been proposed to collaborate devices to train a shared Machine Learning model in a distributed way while keeping the training data locally [2].

The BCFL can avoid the single point of failure and targeted attacks [3]. However, the BCFL still presents challenges. The BCFL cannot detect local or aggregated model attacks that malicious devices send [4]. Shayan et al. [5] and Liu et al. [6] respectively proposed the "Proof of Federated" consensus and secure smart contracts to resist model update attacks. However, these schemes require a large amount of public test data as support, and lack defenses against model aggregation attacks. Additionally, due to the training devices being different and unknown [7], selecting improper devices from a large number of candidate devices will lead to inefficient training of BCFL. Lu et al. [8] and Kang et al. [9] respectively proposed deep reinforcement learning and a reliable node selection scheme to select devices based on the local training situation. But these schemes either assume that devices qualities are known in advance or cannot ensure devices to report information honestly. Therefore, the BCFL needs an unknown device selection algorithm to provide high-quality local models and efficient training.

Based on the above problems, we propose a three-phase consensus algorithm called "Proof of Model Quality" (PoMQ) to resist two kinds of Byzantine model attacks. In the model aggregation phase of PoMQ, we combine the FedAvg [10] and Multi-Krum [11] algorithms, which resist model update attacks. In the verifying and voting phase, the PoMQ combines the model verification problem, and uses the PoW solution mode and the PBFT voting process to resist model aggregation attacks. To improve the training efficiency of BCFL, we define the unknown device selection problem as a Combinatorial Multi-Armed Bandit problem (CMAB) to maximize the total device quality under limited training rounds. We also design an online learning algorithm based on PoMQ consensus (OLAC) to solve the CMAB problem. The main contributions are summarized as follows.

- We propose the PoMQ consensus algorithm to defend against two kinds of Byzantine model attacks and improve the robustness of BCFL. The PoMQ combines the FedAvg and Multi-Krum algorithms to resist model update attacks in the model aggregation phase. The PoMQ also uses the PoW solution mode and the PBFT voting process to resist model aggregation attacks in the verifying and voting phase.
- We define the unknown device selection problem as CMAB to maximize the total device quality and then improve the training efficiency of BCFL. We also propose the OLAC algorithm to solve the CMAB problem and realize the adaptive selection of unknown candidate devices.
- We establish the BCFL system prototype based on Hyperledger Fabric to implement the PoMQ consensus and OLAC algorithm. The extensive evalu-

ation results show that PoMQ and OLAC can better improve the robustness and training efficiency of BCFL.

## 2 Related Works

In recent years, building a secure BCFL has become a popular direction in the research field. Chen et al. [12] proposed "proof of Validation" consensus algorithm (PoV) to ensure that local update models are effective. A reliable test dataset and an accuracy threshold are stored on the blockchain. Miners utilize this dataset to validate updates. Shayan et al. [5] proposed a "Proof of Federated" (PoF) consensus, and using consistent hashing and verifiable random function to select key roles. Liu et al. [6] and Lu et al. [13] respectively proposed a secure smart contracts and "Proof of Training Quality" consensus to defend against model attacks. However, these systems lack protection against model aggregation attacks, and ignore public data resource limitations. Additionally, due to the storage and computing resources of unknown devices being quite different, the BCFL's training efficiency is generally unstable and inefficient. Therefore, It is significant to realize an efficient unknown device selection algorithm in BCFL.

Kang et al. [9] proposed a reliable node selection scheme for BCFL. According to nodes' historical behaviour and training time, information is converted into their respective reputation values through the multi-weight subjective logical model. Xiong et al. [7] used the negative interaction influence and the interaction frequency between nodes and task publishers to construct the reputation value. Lu et al. [8] used deep reinforcement learning for node selection. According to train local model time required, the accuracy of the model and the distance between the vehicle and the roadside node, system chooses the suitable device to improve learning efficiency. Current research on device selection schemes either requires honest reporting of training information or a trusted central node is required to publish the device selection scheme. There is a lack of unknown device selection for BCFL, which is crucial for improving training efficiency.

## 3 BCFL System

We explore a blockchain-driven distributed learning scenario called BCFL to realize decentralized federated learning (DFL) between devices and edge nodes. The architecture of the BCFL is shown in Fig. 1, which has three-layer network architecture consisting of application layer, blockchain layer and device layer. In the device layer, devices such as sensors, cameras, vehicles, smart appliances, etc., participate in DFL tasks and are responsible for collecting data, training local model and publishing local update models into the blockchain network through transaction $tx$. In the blockchain layer, A large number of edge nodes jointly build the blockchain network, which store DFL models, establish PoMQ consensus and OLAC algorithm to realize the training process of DFL. In the application layer, the task publisher publishes DFL tasks from the blockchain network and utilizes the interface provided by blockchain to obtain global models.

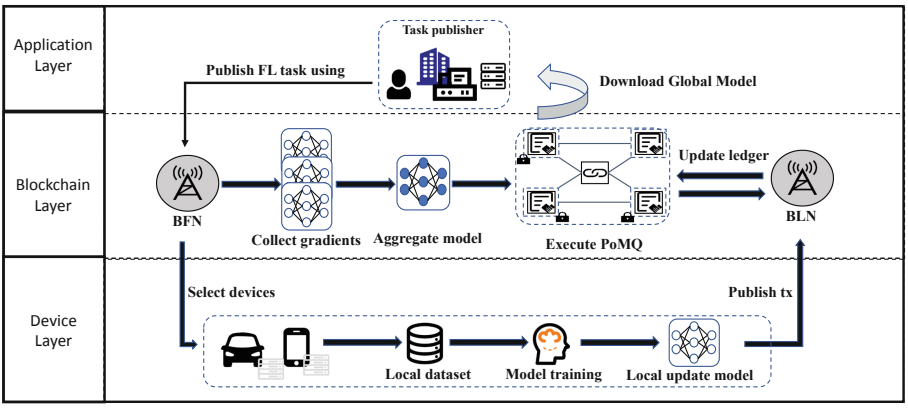



**Fig. 1.** The overall architecture of BCFL

In order to describe the whole system in more detail, We divide the roles of edge nodes into BCFL follower node(BFN) responsible for aggregating the local update model, and the BCFL leader node(BLN) is responsible for generating new blocks. (1) BFN. All edge nodes are BFNs, which perform model verification on transactions (local update models) published by devices. BFN also aggregates valid models. In addition, as a blockchain node, BFN needs to achieve PoMQ consensus through voting and election. And as a candidate, BFN competes for the right to generate blocks and completes to update the ledger. (2) BLN. In each PoMQ consensus, all BFNs will vote to select a BFN to become the BLN by competing, according to the aggregation model from BFNs.

## 4 Proof of Model Quality

In the DFL process, two types of Byzantine model attacks initiated by malicious devices will challenge the security of BCFL system. Therefore, we propose a novel three-phase consensus algorithm PoMQ to defend against model attacks. Overall, The PoMQ is divided into three phases which will be described in detail next.

### 4.1 Model Aggregation Phase

There is a type of model attack in DFL, namely "model update attacks". Malicious devices mislabel the local data into a different class, causing local model to misclassify it, which leads to serious impairment of global model convergence. We combine the FedAvg with Multi-Krum algorithm in model aggregation phase, which can effectively verify the local update models submitted by each device.

For each round of PoMQ, We assume that the device set $D = \{d_1, d_2, \ldots, d_n\}$ is selected to participate in the DFL task. The datasets are $\{D_1,\ D_2, \cdots,\ D_n\}$. Without loss of generality, we assume that each training sample in a dataset is a

set of input-output pairs $(x,\ y)$, where $x$ is feature and $y$ is the label. The model parameters of $t-th$ round are denoted as $w_t$. For each sample $k$, the loss function is defined as $f_k(w) = l\left(x_k, y_k \mid w\right)$. Therefore, the loss function of Device $i$ on the mini-batch $b_i$, a randomly sampled subset from $D_i$, can be written as $f_{b_i}(w)$. The goal of Device $i$ is to minimize the loss on each mini-batch:

$$\min F_i(w) = \mathbb{E}_{b_i \sim D_i} f_{b_i}(w). \tag{1}$$

By applying the gradient descent algorithm on the mini-batch, the local model of Device $i$ can be updated according to:

$$w_i \leftarrow w_t - \mu_i \nabla f_{b_i}\left(w_i\right), \tag{2}$$

where $\mu_i$ is the learning rate of this device. After Device $i$ performs $E$ epochs of local model training using the local dataset $D_i$, $w_i$ is published to the blockchain network through $tx$.

When $\mathrm{BFN_i}$ receives enough $tx$, $\mathrm{BFN_i}$ will use the Multi-Krum algorithm to filter out abnormal models and select a valid set of local update models. The specific process is as follows:

(1) $\mathrm{BFN_i}$ will calculate the score $s(i)$ of each Device $i$, where $s(i)$ represents the sum of the Euclidean distances between local update model parameters after Device $i$ and other devices $j$ are updated in different gradient directions. $s(i)$ is shown in Eq. 3:

$$s(i) = \sum_{j=D \notin d_i} \left\| \left( w_t - \sum_{E} \mu_i \nabla f_{b_i}\left(w_i\right) \right) - \left( w_t - \sum_{E} \mu_j \nabla f_{b_j}\left(w_j\right) \right) \right\|^2. \tag{3}$$

(2) The Multi-Krum algorithm can guarantee against m Byzantine devices in the BCFL with $n$ devices (where $2m+2 < n$). Therefore, $\mathrm{BFN_i}$ will select $n-m$ local update models with the most miniature scores to aggregate and eliminate the remaining models. The model aggregation is shown in Eq. 4:

$$w_{t+1} = \frac{1}{n-m} \sum_{i=1}^{w_{verified}} w_i, \tag{4}$$

where $w_{verified}$ means the valid local update model, $w_{t+1}$ means aggregate model parameters calculated from $\mathrm{BFN_i}$ in the $t-th$ round. Then $\mathrm{BFN_i}$ sends the aggregation result $M_{vote}$ to $\mathrm{BFN/BFN_i}$, waiting for the verification and voting of other nodes. Regardless of the different network latency, $\mathrm{BFN_i}$ stores all $M_{vote}$ from other BFNs in "local validation pool" $R$. All model aggregation results in this round are stored in $R$. Due to the uncertainty of the calculation process, the size of $R$ is also not fixed.

### 4.2 Verifying and Voting Phase

The BCFL selects one or more edge nodes to aggregate local update model. However, system cannot guarantee these edge nodes' credibility and reliability, which

may aggregate local update models submitted by malicious devices. To deal with the above problem, in Verifying and Voting Phase, $\mathrm{BFN_i}$ verifies $M_{vote}$ from $R$ and votes the optimal result. All BFNs elect the optimal aggregation model parameters $W_{opt}$, according to the voting results. The $\mathrm{BFN_i}$ which calculates the $W_{opt}$ will be the BLN in this round. The specific process is as follows:

(1) All $\mathrm{BFN} = (\mathrm{BFN_1}, ..., \mathrm{BFN_i}, ...\mathrm{BFN_j})$ respectively execute local model aggregation scheme. For example, $\mathrm{BFN_i}$ verifies and obtains the aggregate model parameters set from $W = (w_1, \ldots, w_j)$ in $R$. The corresponding valid device ID set $D = (D_1, \ldots, D_j)$ is also obtained. Based on these two indicators, $\mathrm{BFN_i}$ evaluate each aggregation model scheme, and calculate the optimal aggregation model parameters $W_{opt}$, as shown in Eq. 5:

$$W_{opt} = \min \left( \sum_{k \in R} \|w_1 - w_k\|^2 + \alpha \frac{D_i - D_1}{D}, \ldots, \sum_{k \in R} \|w_j - w_k\|^2 + \alpha \frac{D_i - D_j}{D} \right), \tag{5}$$

where $\alpha$ is a hyperparameter that measures the validity device set's effective to the aggregated model.

(2) After obtaining $W_{opt}$, $\mathrm{BFN_i}$ will generate $v_i$ which combine the $W_{opt}$ and its own identity signature. $v_i$ is broadcasted to $\mathrm{BFN/BFN_i}$, supporting $BFN_{opt}$ become the BLN. Then, $\mathrm{BFN/BFN_i}$ need to authenticate $v_i$ from $\mathrm{BFN_i}$.

(3) $\mathrm{BFN_i}$ verifies all the $v$, counts the number of valid votes and saves the voting results. If $\mathrm{BFN_i}$ find that it has obtained the most amount of support votes, it will send a broadcast requesting to become the BLN to $\mathrm{BFN/BFN_i}$. $\mathrm{BFN_i}$ compare with the voting results. If more than half of the BFNs voting results are consistent with $\mathrm{BFN_i}$. $\mathrm{BFN_i}$ becomes BLN and is responsible for generating a new block.

### 4.3 Ledger Update Phase

BLN needs to encapsulate all model information generated by this round into a new block body. Then the $W_{opt}$ will be encapsulated into the block header. BLN broadcasts $Block_{new}$ to all BFNs, and BFN will verify the $Block_{new}$. If the $Block_{new}$ is valid, BFN update the local blockchain ledger. At this point, a complete DFL training process is over.

## 5 Unknown Device Selection Algorithm

In each round of PoMQ consensus, the BCFL system needs to select some devices from a large number of unknown candidate devices to participate in DFL training. Due to the training resources of devices being pretty different, improper selection of unknown devices will lead to inefficient training. To solve the efficiency problem of the BCFL, we define the unknown device selection problem as a CMAB problem to maximize the total device quality under limited training rounds, and propose the OLAC algorithm to solve CMAB. To better describe OLAC, some concepts that need to be defined as follows:

Definition 1 (Round): A round of device selection in BCFL, expressed as $r \in \{1, 2, \ldots\}$. It should be noted that there will be multiple rounds of PoMQ consensus process after device selection.

Definition 2(Unknown Device): BCFL exists a set containing $N$ unknown candidate devices, expressed as $\mathcal{N} = \{1, 2, \ldots, N\}$.

Definition 3(Device Quality): After performing multiple rounds of PoMQ consensus, each selected Device $i$ will get the device quality, which is expressed as $q_{i,r} \in [0, 1]$. Each $q_{i,r}$ follows an unknown distribution with an unknown expectation $q_i$, which also indicates that the Device $i$ is unknown. It is assumed that $q_i$ only depends on the computing resources, data resources and communication resources, and is calculated according to Device $i$'s local update model.

### 5.1 Device Quality Model

We define the unknown device selection problem as a CMAB problem, where each device in $\mathcal{N}$ represents an arm. Device quality was considered as the reward for pulling the corresponding arm. Each round of device selection will pull $k$ arms, and pulling the $i-th$ arm means that Device $i$ is selected to participate in the DFL task. A feasible bandit strategy needs to be determined in CMAB to maximize the total expected revenue (total device quality), thereby improving the training efficiency of DFL. The bandit strategy and total revenue are defined as follows:

$$\phi = \{\phi_1, \phi_2, \ldots, \phi_r, \ldots\}, \tag{6}$$

$$\phi_r = (\phi_{1,r}, \phi_{2,r}, \ldots, \phi_{N,r}), \tag{7}$$

Eq. 6 represents a series of bandit strategy, Eq. 7 represents a round of bandit strategy. $\phi_{i,r} \in \{0, 1\}$ indicates whether Device $i$ is selected in the $r$ round.

$$Maximize: \ E[R(\phi)] = \sum_{r} \sum_{i=1}^{N} q_{i,r}\phi_{i,r}, \tag{8}$$

$$subject\ to: \ \sum_{i=1}^{N} \phi_{i,r} = K, \forall r,$$
$$\phi_{i,r} \in \{0, 1\}, \ \forall\ i \in \mathcal{N}, \forall r,$$

Equation 8 represents the total expected revenue, which refers to the total selected device quality. The goal is to maximize the total expected revenue with limited training rounds. Equation 8 is also restricted that the number of devices to be selected in each $r$ should be $K$, and all bandit strategies to be binary.

A fair metric to evaluate $q_{i,r}$ is crucial. We construct a device quality model based on whether Device $i$'s local update model is used for model aggregation in the PoMQ consensus. $q_{i,r}$ can be expressed as a meta-vector $\{h_{i,r}, l_{i,r}, u_{i,r}\}$,

which represent the high-quality model probability, low-quality model probability, and uncertain probability, where $h_{i,r} + l_{i,r} + u_{i,r} = 1$, $h_{i,r}, l_{i,r}, u_{i,r} \in [0, 1]$. Those are denoted as:

$$h_{i,r} = (1 - u_{i,r}) \frac{\alpha_{i,r}}{\alpha_{i,r} + \beta_{i,r}}, \tag{9}$$

$$l_{i,r} = (1 - u_{i,r}) \frac{\beta_{i,r}}{\alpha_{i,r} + \beta_{i,r}}, \tag{10}$$

$$u_{i,r} = 1 - s_{i,r}, \tag{11}$$

where $\alpha_{i,r}$ represents the times that Device $i$'s local update models are aggregated in round $r$. $\beta_{i,r}$ represents the times that SLN eliminates Device $i$'s local update models. $s_{i,r}$ represents the times of successful transaction publication. This represents the quality of the communication, which will affect the uncertainty of the device. According to $\{h_{i,r}, l_{i,r}, u_{i,r}\}$, Device $i$'s quality model in round $r$ can be obtained as shown in Eq. 12:

$$q_{i,r} = h_{i,r} + a u_{i,r}, \tag{12}$$

where $a \in [0, 1]$ represents the influence of uncertainty probability.

### 5.2 Online Learning Algorithm Based on PoMQ Consensus(OLAC)

We design the OLAC algorithm to solve the CMAB problem. OLAC is divided into two phases: exploration and exploitation. In the exploration phase, candidate devices are uniformly selected for DFL training, and based on PoMQ consensus to learn device's quality. In the exploitation stage, a greedy strategy based on the upper Confidence Bound (UCB) is adopted to select high-quality devices. The OLAC's algorithm details will be introduced in detail below.

(1) Exploration stage: The smart contract $\text{Contrast}_{\text{sel}}$ selects devices to learn their expected device quality. Since each device's quality is unknown, system needs to treat equally. OLAC explores all candidate devices in a round-robin method. This means selecting the device $\{1, \ldots, K\}$ in the first round, The second round selects the device $\{K + 1, \ldots, 2K\}$, and so on. At the same time, Keep two vectors $n_r = (n_{1,r}, \ldots, n_{N,r})$ and $\hat{q}_r = (\hat{q}_{1,r}, \ldots, \hat{q}_{N,r})$ as empirical knowledge learned from PoMQ consensus. Specifically, $n_{i,r}$ and $\hat{q}_{i,r}$ represent the number of Device $i$'s quality learned by system and Device $i$'s quality sample mean respectively. Before the next round of device selection, the SLN uploads the local model aggregation results to $\text{Contrast}_{\text{sel}}$, which updates device's quality information. Once Device $i$ is recruited, the corresponding device quality will be learned one time. Therefore, $n_{1,r}$ is updated as shown in Eq. 13:

$$n_{i,r} = \begin{cases} n_{i,r-1} + 1, & \phi_{i,r} = 1 \\ n_{i,r-1}, & \phi_{i,r} = 0 \end{cases}. \tag{13}$$

Device $i$'s quality sample mean is updated as shown in Eq. 14:

$$\hat{q}_{i,r} = \begin{cases} \frac{\hat{q}_{i,r-1} n_{i,r-1} + q_{i,r}}{n_{i,r-1} + 1}, & \phi_{i,r} = 1 \\ \hat{q}_{i,r-1}, & \phi_{i,r} = 0 \end{cases}. \tag{14}$$

The device quality information learned in the exploration phase will be used for device selection in the exploitation phase. Considering the uncertainty of device quality assessment, instead of using the quality sample mean to evaluate, We introduce a new vector $\hat{q}^+ = \left(\hat{q}_1^+, \cdots, \hat{q}_N^+\right)$ to represent the UCB index of each device. By the end of $r$ round, $\hat{q}_i^+$ is shown in Eq. 15:

$$\hat{q}_i^+ = \hat{q}_{i,r} + \sqrt{\frac{\delta \cdot \ln\left(\sum_{i' \in \mathcal{N}} n_{i',r}\right)}{n_{i,r}}}, \tag{15}$$

where $\delta$ is a positive hyperparameter that brings flexibility to the OLAC.

(2) Exploitation phase: Based on the UCB index, OLAC can determine selected devices set in the exploitation phase. Since the device selection in the exploitation stage can be regarded as a series of 0–1 knapsack problems, Therefore, $\text{Contrast}_{\text{sel}}$ adopt the greedy strategy based on ucb to select high-quality devices, and the device with the highest $\hat{q}^+$ will be selected in each round. OLAC first calculates $\hat{q}_i^+$ for each device, then sorts all devices into the set $S = (s_1, s_2, \ldots, s_N)$, and guarantee $\hat{q}_{s_1}^+ \geq \cdots \geq \hat{q}_{s_N}^+$. Then the optimal K devices are selected to participate in DFL based on a greedy strategy. When required training rounds are reached, the exploitation phase will stop.

## 6 Implementation

We detail the practical deployment of BCFL, which includes PyTorch 1.9 based DFL training process and PoMQ consensus, Hyperledger Fabric 2.1 (Fabric) based block chain network. By building on the general-purpose API in PyTorch, the BCFL can support any model that can be optimized using SGD. Fabric is a permissioned distributed ledger technology platform, which is suitable for the consortium settings of BCFL. With the Public Key Infrastructure (PKI)-based membership management, the Fabric network has plentiful capabilities to control the access of devices. The experiment deploys a complete Fabric-based blockchain network in a PC, which contains three BFN nodes. In addition, the training process of devices is also performed in the same PC.

To implement the OLAC algorithm in the BCFL system, we use Go 1.14 to implement the smart contract sel_device.go in Fabric. sel_device.go deploys the complete unknown device selection process. Devices call the interface query() of sel_device.go to judge whether to participate in this round of PoMQ consensus. SLN calls the interface update() to update the quality information of all candidate devices.

## 7 Experiment Evaluation

Experiments evaluate BCFL from two performance indicators: robustness and training efficiency. The experiments use the MNIST image dataset [14]. The LeNet-5 is used as the DFL model, which consists of two convolutional layers with maximum pooling and three fully connected layers. Each device trains LeNet-5 with a batch size of 10, a learning rate of 0.01, and a number of local iterations of 5. Cross-entropy is used as a loss function to modify the training parameters. The MNIST was equally divided into 100 groups for each divice. Each device is assigned a group which has 600 training samples. The BCFL contains 100 candidate devices. In each round of PoMQ, the BCFL will select 20 devices to participate in the DFL training process from 100 candidate devices.

In the robustness experiments, two baselines will be compared with PoMQ: (1) FL with FedAvg (FedAvg) [10] consists of a server responsible for model aggregation and devices responsible for local model training. (2) Biscotti [5] is a DFL system, where the PoF consensus maintains edge nodes' state consistency.

In the training efficiency experiments, OLAC's exploration phase accounts for 30% of the total training rounds. OLAC will compare with three algorithms: (1) The Reputation-based Worked Selection Scheme (RWS) [7] calculates each device's reputation value through a multi-weight subjective logic model according to the historical behavior in DFL. (2) The $\alpha$-optimal algorithm has full knowledge about devices qualities and always recruits the optimal K workers. (3) The Random algorithm randomly selects K workers in each round. The experiments will use the above algorithm to select candidate devices, which perform the PoMQ consensus for DFL training.

### 7.1 Evaluation Results

To evaluate system's robustness, We first simulate malicious devices' model update attack by modifying number 1's label to 2 and number 2's label to 3. The proportion of malicious devices $M_d$ in the 20 training devices was adjusted to 20% and 30%. The experiments are shown in Fig. 2 (a). The global model of FedAvg is difficult to converge. Although Biscotti are influenced in the early rounds, the global model can still converge as the amount of training data increases. BCFL can always maintain a stable convergence rate, and is still faster than Biscotti.

Figure 2 (b) simulates model aggregation attacks by aggregating local update model published by malicious devices. Experiment adds 10% and 20% devices to perform model aggregation attacks based on the 10% model update attack. It is difficult for Biscotti to maintain a stable convergence rate. Although Biscotti prevents model aggregation attacks by allocating shares to each edge node, which is adjusted dynamically during the training process. This delayed penalty will affect global model's training progress in the early rounds. DFL is very sensitive to any model aggregation attack. With the verification and voting phase of PoMQ, the impact of model aggregation attacks is eliminated.

The experiment evaluates the OLAC algorithm by observing total revenue and Acc, where the total training round $R_{total}$ is 50. The result is shown in

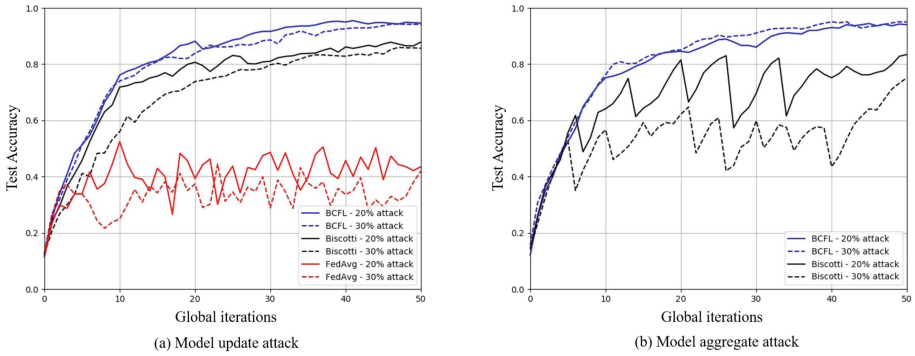



**Fig. 2.** (a) The impact of model update attacks on global model's test accuracy. (b) The impact of model aggregation attacks on global model's test accuracy

Fig. 3 (a). The OLAC's total revenue is at least 23% higher than RWS, almost two times that of the Random algorithm. In fact, the total revenue of OLAC is even going to catch up with that of the $\alpha$-optimal algorithm which knows the quality information in advance. We can also find that Acc has a similar pattern to the total revenue. The experiment also proves that the total revenue is positively correlated with Acc. This is because the participation of high-quality devices will bring better data resources, computing resources, and faster model convergence.

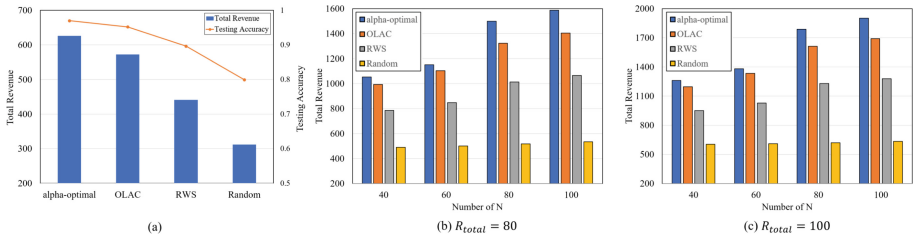



**Fig. 3.** (a) Performance Comparison of Unknown Device Selection Algorithms. (b) and (c) The effect of N on total revenue

The experiment evaluates the OLAC algorithm influenced by the number of candidate devices N. We fix the training round $R_{total}$, and the scale of candidate devices is expanded from 40 to 100 in increments of 20. The results are shown in Fig. 3 (b) and (c). It can be observed that the total revenue of the OLAC algorithm increases with N. Furthermore, the OLAC algorithm can always obtain higher than that of the RWS and Random algorithm. This is because when the values of $R_{total}$ and $K$ are fixed and more high-quality devices appear, which will lead to better choices. When N is much larger than K, the learning ability of OLAC in the exploration phase will be reduced. Therefore, the OLAC algorithm has the largest gap compared to that of $\alpha$-optimal with N being 100.

## 8 Conclusion

In this paper, we explore a BCFL scenario, and propose the PoMQ consensus algorithm, which combines the FedAvg and Multi-Krum algorithms to defend against model update attacks, and is based on the model verification process to resist model aggregation attacks. Moreover, we model the unknown device selection problem as a CMAB, and propose the OLAC algorithm to solve CMAB, which improves DFL's training efficiency. Finally, we establish the BCFL system prototype. Extensive experiments show that our solution provides stronger robustness and training efficiency for BCFL.

## References

1. Wang, J., Feng, X., Xu, T., Ning, H., Qiu, T.: Blockchain-based model for nondeterministic crowdsensing strategy with vehicular team cooperation. IEEE Internet Things J. **7**(9), 8090–8098 (2020)
2. Nguyen, D.C., et al.: Federated learning meets blockchain in edge computing: opportunities and challenges. IEEE Internet Things J. **8**(16), 12806–12825 (2021)
3. Otoum, S., Ridhawi, I.A., Mouftah, H.T.: Securing critical IOT infrastructures with blockchain-supported federated learning. IEEE Internet Things J. **9**(4), 2592–2601 (2022)
4. Fung, C., Yoon, C.J.M., Beschastnikh, I.: The limitations of federated learning in sybil settings. In: 23rd International Symposium on Research in Attacks. Intrusions and Defenses, pp. 301–316. USENIX, San Sebastian, Spain (2020)
5. Shayan, M., Fung, C., Yoon, C.J.M., Beschastnikh, I.: Biscotti: a blockchain system for private and secure federated learning. IEEE Trans. Parallel Distrib. Syst. **32**(7), 1513–1525 (2021)
6. Liu, Y., Peng, J., Kang, J., Iliyasu, A.M., Niyato, D., El-Latif, A.A.A.: A secure federated learning framework for 5G networks. IEEE Wirel. Commun. **27**(4), 24–31 (2020)
7. Kang, J., Xiong, Z., Niyato, D., Xie, S., Zhang, J.: Incentive mechanism for reliable federated learning: a joint optimization approach to combining reputation and contract theory. IEEE Internet Things J. **6**(6), 10700–10714 (2019)
8. Lu, Y., Huang, X., Zhang, K., Maharjan, S., Zhang, Y.: Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. IEEE Trans. Veh. Technol. **69**(4), 4298–4311 (2020)
9. Kang, J., Xiong, Z., Niyato, D., Zou, Y., Zhang, Y., Guizani, M.: Reliable federated learning for mobile networks. IEEE Wirel. Commun. **27**(2), 72–80 (2020)
10. McMahan, B., Moore, E., Ramage, D., Hampson, S.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, pp. 1273–1282. PMLR, Fort Lauderdale, USA (2017)
11. Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: byzantine tolerant gradient descent. Adv. Neural. Inf. Process. Syst. **30**(4), 119–129 (2017)
12. Chen, L., Chiu, T., Pang, A., Cheng, L.: Fedequal: Defending model poisoning attacks in heterogeneous federated learning. In: IEEE Global Communications Conference, pp. 1–6. IEEE, Madrid, Spain (2021)

13. Lu, Y., Huang, X., Dai, Y., Maharjan, S., Zhang, Y.: Blockchain and federated learning for privacy-preserved data sharing in industrial Iot. IEEE Trans. Industr. Inf. **16**(6), 4177–4186 (2020)
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)

# TraceDroid: Detecting Android Malware by Trace of Privacy Leakage

Yueqing Wu[1], Hao Fu[2], Guoming Zhang[1], Bin Zhao[1], Minghui Xu[1], Yifei Zou[1], Xiaotao Feng[3], and Pengfei Hu[1(✉)]

[1] Shandong University, Qingdao 266237, CN, China
yueqingwu@sdu.edu.cn, guomingzhang@sdu.edu.cn, zhaobinsdu@sdu.edu.cn, mhxu@sdu.edu.cn, yfzou@sdu.edu.cn, phu@sdu.edu.cn
[2] UC Davis, Davis, CA 95616, US
haofu@ucdavis.edu
[3] JD.com American Technologies Corporation, Mountain View, CA 94043, US
xiaotao.feng@jd.com

**Abstract.** Along with the popularity of the Android operating system, 98% of mobile malware targets Android devices [1], which has become one of the primary source for privacy leakage. Detecting malicious network transmissions in these apps is challenging because the malware hides its behavior and masquerades as benign software to evade detection. In this work, we propose TraceDroid, a framework that can automatically trace abnormally sensitive network transmissions to detect the malware. By leveraging the static and dynamic analysis, the sensitive informations can be firstly inferred from the call graph, and then, the sensitive transmissions can be detected by analyzing the network traffic per transfer and sensitive information with a machine learning classifier. We validate TraceDroid on 1444 malware and 700 benign applications. And our experiments show that TraceDroid can detect 3433 sensitive connections across 2144 apps with an accuracy of 94%.

**Keywords:** Android malware detection · Static analysis · Dynamic analysis

## 1 Introduction

With the widespread use of mobile devices running Android, it has become the dominant operating system. Users can install applications from third parties without performing any malware checks [31]. As a result, the number of malware samples targeted the Android ecosystem has skyrocketed in recent years, which poses significant threat to user's privacy. Bad guys can infer information such as home address from user privacy, causing great harm to users [3,5,6,29].

This work is supported by the National Key Research and Development Program of China (No. 2021YFB3100400), the Shandong Science Fund for Excellent Young Scholars (No. 2022HWYQ-038).


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 466–478, 2022.
https://doi.org/10.1007/978-3-031-19208-1_38

However, detecting malware on mobile devices is a big challenge. Due to the limitations of Android's permission mechanism [23], malware can disguise itself as a benign application by using normal APIs to fool the security audit module [28]. For instance, when a user clicks button of an application which shows sending SMS messages, it might transmit the user's sensitive contact information along with that SMS at the background without notifying the user. Furthermore, the sensitive network transmissions are not necessarily regarded as malicious traffic, which is probably sent to a benign server for normal operation. Hence, the key of differentiating normal and abnormal sensitive transmissions lies in properly understanding the intent. To detect the malware, previous works [18, 19] leverage natural language processing techniques to understand whether an application's description is consistent with its permission setting. However, they are unable to infer the privacy leakage. Some of the detection methods are limited by the complexity of Android APIs and runtimes, which include millions of lines of code [8]. Most importantly, they only focus on detecting sensitive traffic and fail to distinguish between normal and abnormal sensitive traffic. Some people use the network traffic of application for detection, but they are not always accurate [17].

**TraceDroid.** To address the aforementioned issues, we propose TraceDroid, which combines static analysis, dynamic analysis, and machine learning methods to detect abnormal sensitive network traffic induced by malware. First, TraceDroid employs a static analysis approach to derive possible execution traces of application and to identify sensitive information. However, some applications send information after encrypting the name of an external server in order to avoid security checks, and the destination server address is only visible at runtime. Therefore, TraceDroid also utilizes a dynamic analysis method to collect runtime information to uncover disguised malware. This hybrid approach achieves better performance than purely static or dynamic analysis methods. Moreover, we are able to obtain more network traffic data and provide a better characterization of network behavior by leveraging a hybrid analysis approach than the widely used static analysis approach [8,14]. At last, the transmission data will be used for model training. We can apply the well tuned model to identify malware even when the source code is unavailable, which could be directly integrated into network-based intrusion detection system.

Our contributions can be summarized as follows.

- We propose an Android malware detection method, TraceDroid, which performs ingress analysis of malicious traffic by lightweight static analysis and pinpoints negative network transmissions by detailed dynamic analysis.
- TraceDroid can distinguish abnormal network traffic. Compared with previous work, TraceDroid achieves higher accuracy and better runtime efficiency while being more robust to malware variants and Android API updates.
- We evaluate TraceDroid on a dataset which includes 1444 malware from various malware datasets and 700 market apps downloaded from AppStore. The

results show that TraceDroid can detect 3433 sensitive connections across 2144 apps with an accuracy of 94%.

**Paper Organization.** The remaining paper is organized as follows. We introduce related work in Sect. 2. In Sect. 3, we presents our system TraceDroid, and describes its technical details. Section 4 performs the evaluation of our system. Section 5 summarizes our work.

## 2 Related Work

Emerging research efforts have been made on Android malware detection, which can be broadly classified into Signature-Based( [11,13]), ML-Based([16,26]), and Behavior-based([25,26]) respectively. The signature-based methods have low computing complexity and can provide specific evidence (detected malicious feature codes) to explain. However, such methods can easily be bypassed by malicious code. Machine learning-based methods have been investigated to detect Android malware. However, the performance of pure machine learning methods is limited to selected features and existing training datasets [4]. MaMaDroid [16] utilizes the statistical methods of Markov chains to detect malware, but it is also vulnerable to some attacks using evasion techniques [7].

FlowDroid [2] and DroidSafe [10] both utilize static analysis solutions to precisely detect suspicious information flows, but the results become inaccurate because the visited code paths are not always feasible. Based on dynamic analysis tools, TaintDroid [9] is a real-time privacy monitoring system. By modifying the Dalvik virtual machine, TaintDroid can report information leaks while the application runs on the Android device. It only identifies leaks triggered during execution, so a driver with good code coverage is required. Naturally, some tools use hybrid analysis, such as AppAudit, to avoid the weaknesses of using a single analysis. Still, it leaves out most unknown branches and only follows one chapter identified by static checks. The downside of all the tools mentioned above is that they treat any breach of user data as malicious, leading to numerous false alarms. Compared to the above system, TraceDroid has the flexibility to extend code coverage based on context and explore as many possible paths as possible when discovering unknown branches. Our hybrid program analysis approach further improves detection efficiency and reduces false positives.

## 3 Design Of Tracedroid

### 3.1 Overview

This section introduces the design of TraceDroid, which leverages data flow analysis technology to determine the source of perceptual data and then track the information flow to select the final destination of sensitive data. The system overview of TraceDroid is given in Fig. 1. It includes three main components: Static analysis, dynamic analysis, and classification.

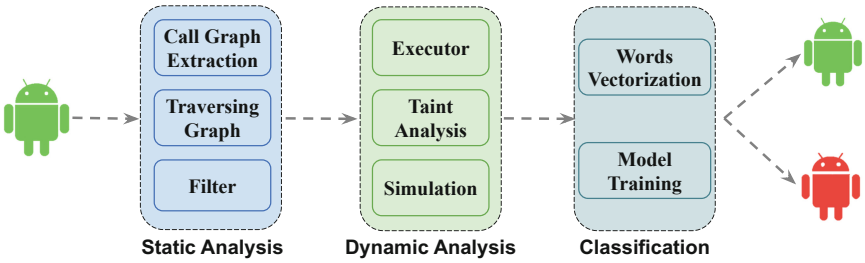



**Fig. 1.** The workflow of TraceDroid.

### 3.2 Static Analysis

The goal of static analysis is to construct the call graph of the target application, which can assist the dynamic analysis and improve its efficiency.

**Call Graph Extractions.** In contrast to traditional Java programs, which have only one entry point (i.e. main), Android apps have multiple entry points. Specifically, Android applications consist of multiple components that each Activity or Service component is a Java class, and each event listener and lifecycle method serve as an entry point for a specific event. Thus, to fully capture the sensitive information traces, all possible transitions in the application's lifecycle must be captured precisely.

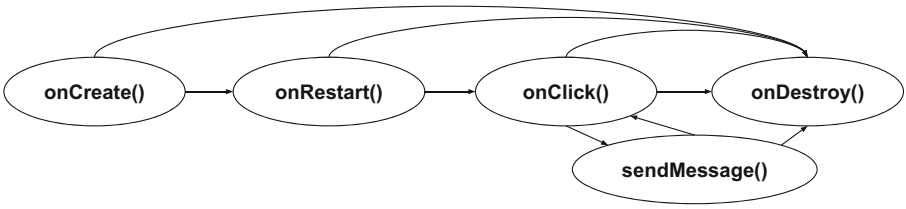



**Fig. 2.** An example of call graph

In order to construct an application's call graph, prior work typically creates one or more virtual main routines that are shared by multiple components [2,27]. However, some components without leaking information will be included with the above methods which will introduce unnecessary interference. Besides, the shared virtual main program may obscure the connection between components. Instead of building a shared virtual main program, each component in TraceDroid has a separate call graph to eliminate clutter and reduce the overhead of dynamic analysis. As shown in Fig. 2, event listeners `onClick()` are embedded in the component and registered after `onCreate()`. `onClick()` is a UI callback function that will be called when the appropriate button is clicked. As the underlying

static analysis framework, we leverage Soot [24] which is a Java optimization and analysis framework to extract call graphs.

**Traversing Graph.** After obtaining the call graph, TraceDoid leverages it to quickly locate the sensitive APIs call. A `source` is an API call which accesses to the sensitive information. Sensitive information includes device identifiers, SMS, contact data, etc. All of these data items are retrieved, sent, or stored through the Android APIs, which is listed in Susi [21]. Typically, `getText()` at line 8 shown in Listing 1.1 is a source. For each `source`, the corresponding entry point of the component is extracted by applying a graph traversal algorithm in the call graph. For instance, the entry point `onRestart()` of the component `PrivataDateLeakag` in Listing 1.1 is located through breadth-first search beginning with `getText()` on the call graph.

**Filter.** Since the static analysis component has a lot of unnecessary entry functions, we thus use filters in the static analysis results to improve the efficiency of our analysis. Some keywords are used in the filter to filter out unnecessary entries such as those containing Android kernel features.

### 3.3 Dynamic Analysis

The dynamic analysis component consists of an execution system with a taint analysis module and a simulation of the Android runtime.

**Executor.** The executor is based on a specially designed Dalvik virtual machine which can unpack Android package files and execute bytecode instructions directly. After static analysis, a set of **traces** can be derived. Then the **traces** are fed into the execution system. Note that each trace is a sequence of specific API calls beginning with a lifecycle callback and ending with an API call related to sensitive information.

For instance, for the entry point `onRestart()` in `PrivateDataLeakage`, TraceDroid builds an execution trace `onRestart()` to `onClick()` that informs the executor to invoke `onClick()` after calling `onRestart()`. When the framework restarts the application, the application reads the password from the text box (line 8). When the user clicks the active button (`onClick()`), the password is sent via SMS (line 20). This constitutes a tainted data flow from the password field (**source**) to the SMS API (**sink**). In this example, `sendMessage()` is associated with a button in the application UI, which is triggered when the user clicks the button. The execution trace is generated by applying depth-first search to find a path from `onRestart()` to `onClick()` in the call graph (Fig. 2). The default values of global variables are normally initialized at the lifecycle callbacks such as `onCreate()` and `onStart()`. We choose to perform these callbacks to reduce the unknown variables. AS it reduces the number of unknown branches to be explored, it improves the efficiency of dynamic analysis.

**Listing 1.1.** Example Android Application

```
class PrivateDataLeakage extends Activity {
  private User user = null;
  void onCreate() {...//initiate the activity }
  void onRestart() {
      EditText usernameText = (EditText)findViewById(R.id.username);
          EditText passwordText =
              (EditText)findViewById(R.id.password);
          String uname = usernameText.toString();
          String pwd = passwordText.getText().toString(); //source
          user = new User(uname, pwd);
  }
  void sendMessage(View view) {
      if(user != null){
          String password = getPassword();
          String obfuscatedUsername = "";
          for(char c : password.toCharArray())
          obfuscatedUsername += c + "_";
            String message = "User: " + user.getUsername() + " |
                Pwd: " + obfuscatedUsername;
            SmsManager smsmanager = SmsManager.getDefault();
            Log.i("TEST", "sendSMS"); //sink
            smsmanager.sendTextMessage("+49 1234", null, message,
                null, null); //sink, leak
          }
  }
  void onDestroy() {... //finish the activity}
}
```

**Taint Analysis.** Here, the unknown variables are often closely related to some factors, e.g., user input, device status, surrounding environment, etc. Thus, malicious applications may take advantage of some factors to hide their behavior, creating malicious code that can only be triggered under certain circumstances. To tackle this problem, TraceDroid not only introduces the function of snapshots to handle different cases of unknown quantities, but also proposes a rule base to deal with the problem of path explosion caused by unknown variables. Specifically, if an unknown branch is encountered, TraceDroid creates a snapshot to store the state of the executor and presses the snapshot onto the stack. If the termination condition of a loop or recursion is an unknown quantity, code that contains the loop or recursion may cause an infinite number of paths to be explored. We choose the way that execute the block under the loop only once, and mark all the variables in the block that accept the new value. After exploring the block, the marked variable is symbolically modeled for the rest of the execution. During execution, whenever the source API is invoked, the pollution

analysis module begins to track the propagation of the correspondingly sensitive values. When one or more sensitive values reach a network connection API call (a **sink**, such as the `openConnection()` or `connect()` ), this means that the transport is sensitive, the corresponding runtime information, such as network traffic data, is recorded. We employ the general contamination strategy which has been used in the previous work [9,27] to specify the propagation process.

**Simulation of the Android runtime.** Accurate modeling of the Android runtime state is required to perform taint analysis correctly. Therefore, we manually pad the incomplete Android SDK and emulate the core functionalities. Our inspiration for emulating Android comes from [10]. But the Android device implementation used is only developed for static analysis and does not extend well enough to support our dynamic analysis. We supplemented the android framework to make it support more functions. Meanwhile, the return value of the function is simulated to support our dynamic analysis.

### 3.4 Transmission Classification

The final step is to detect the Android malware by analyzing the traffic generated by the dynamic analysis component. TraceDroid uses a supervised learning approach to train classifiers that we aim to operate in local-based or network-based intrusion detection systems. To generate the representative features from URL sets in the traffic, we finally choose lexical features, as the lexical features contain the purpose of transmissions which can be used to distinguish suspicious and benign traces.

**Words Vectorization.** To extract the lexical features from traffic, a bag-of-words model [15] is employed, which is often used for spam detection. In our framework, URLs can be divided into tags using certain characters as delimiters. Each different tag is then treated as an independent feature. Each collected data stream is converted into a vector of binary values. To reduce the computation cost, we can't use word bags directly because this can result in vast feature Spaces. As described in [22], we limit the size of a feature set by removing tokens that are rarely present in a stream.

**Model Training.** Since TraceDroid is commonly used in traffic classification, we consider Decision Tree as a learning classifier [20,22]. We use labeled transmissions as training and test data, and use ten-fold cross validation [12], which is the standard method for evaluating machine learning solutions. According to the hybrid analysis tool, the traffic generated from different code paths in target app probably goes to the same URL, thus, we merge the same URL connections into one transmission. Later, for the collected transports, we check the target hostname to see if it belongs to an Advertising(AD) server or a malicious server and flag the transports as illegal. Then, we need to check the plain text content

passed through the stream to prevent the server from sending a response that is related to the user data being sent. In order to test the transmission of flow is legal, we will intercept these flows in some special way and repackage these applications. If the function of these applications is affected, then we will mark the transmission of flow as legal and if the application is not affected, so we will have sufficient reason to mark the transmission of flow as illegal transfer.

## 4 Evaluation

### 4.1 Experimental Settings

**Datasets.** We first extract 1223 malicious sensitive transmissions and record the corresponding traffic from the classic malicious software set [30]. In addition, we also obtain 700 malicious samples from `VirusShare`[1], as well as 1147 malicious sensitive transmissions. We crawl 700 applications on the legitimate app store. Since the architecture of android system has changed in recent years, our dataset includes new versions of android applications to make our analysis more convincing.

**Matrics.** In the experiments, we employ the standard F-measure metric, *Accuracy*, $TP, FP, FN$ to evaluate the performance of TraceDroid under different settings. The *Accuracy* refers to the ratio of correctly predicted samples to the total samples. $TP$ denotes the number of correctly classifying normal samples as normal, and $FP$ and $FN$ indicate the number of samples mistakenly identified as malicious and benign respectively.

### 4.2 Comparison with Benchmark

We first evaluate the performance of TraceDroid on base datasets compared with the other two state-of-the-art Android detection frameworks. Table 1 summarizes the detection results on DroidBench. DroidBench[2] is an open-source benchmarking suite that contains 118 hand-crafted applications. Particularly, it can utilize various features of programming languages to bypass static pollution analysis. We removed 14 apps due to the inter-app communication involved and other reasons.

**Compare with Other Static Detection Methods.** The detection accuracy of FlowDroid is only 76.8%, that is because it can not effectively analyze the runtime data of app and the modeling of the lifecycle of FlowDroid is imprecise.

[1] https://virusshare.com/.
[2] https://github.com/secure-software-engineering/DroidBench.

**Table 1.** Detection results on DroidBench

| Tools | FP | Accuracy | Precision |
|---|---|---|---|
| FlowDroid [2] | 10 | 76.8% | 70.5% |
| AppAudit [27] | 2 | 50.5% | 91.3% |
| TraceDroid | 0 | 98.3% | 100% |

**Compare with Other Hybrid Analysis Detection Methods.** As we can see that TraceDroid achieves higher detection accuracy than AppAudit. The first underlying reason is that AppAudit chooses to terminate the current execution when it encounters a sink, but satisfying reachability does not imply malicious transfers, as discussed in Sect. 3. The second, AppAudit does not take into account the diversity of unknown variables and keeps hanging on to an unknown branch in one direction, which can reduce the detection accuracy of AppAudit. Furthermore, since Android contains various mechanisms, AppAudit does not support any of these Android features.

In summary, TraceDroid provides a more complete dynamic analysis implementation that not only simulates the behavior of the Android runtime to support various mechanisms, but also tracks communication between multiple components.

### 4.3 Real App

**Trasmission Detection.** In this section, we further present the performance comparison with VirusTotal[3] VirusTotal is a popular website that scans submitted URLs with the latest 68 anti-virus engines. In Fig. 3, *Malware* represents the 1,444 malware apps we collected, *AppStore* represents the 700 apps we download from the AppStore, and *All* represents all the pieces of Malware and apps. It can be observed that the performance of VirusTotal is always inferior than TraceDroid no matter on which apps.

**Table 2.** Classification results in different scenarios

| Scene | Class | Precision | F-measure |
|---|---|---|---|
| Local-based | Illegal | 0.982 | 0.962 |
| Local-based | Legal | 0.872 | 0.905 |
| Network-based | Illegal | 0.918 | 0.924 |
| Network-based | Legal | 0.910 | 0.915 |

[3] https://www.virustotal.com/.

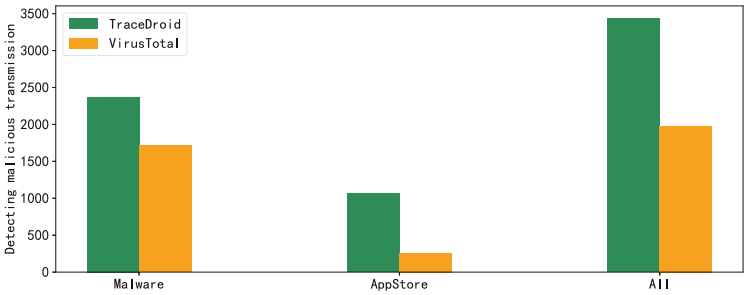



**Fig. 3.** Detecting malicious transmission.

**Comparison with Different Scenarios.** In order to verify the effectiveness of TraceDroid in multiple scenarios, we designed two scenarios: the first is local host system of automatically finds the disclosure points and the second is the scenario involves only the flows of sensitive transmissions. Table 2 shows the classification results in different scenarios. For local-based scene, Table 2 shows that TraceDroid has high precision and F-measure in identifying illegal transmissions. After manually inspecting the misidentified instances, we found that their URLs were very similar to the benign addresses. Also, they put the sensitive data into their body rather than the URL, which makes the URL-based detection more difficult to correctly label them. We plan to consider more features to further reduce the false negatives in the future. For network-based scene, based on the sensitive transmissions we collected, we add the non-sensitive traffic flows to the legitimate class. This reflects the real environment of the network-based detection. Table 2 summarizes our results. As we can see that the prediction accuracy of network-based scenario is slightly lower the local-based detection's.

TraceDroid found 3,433 suspicious behaviors in marketing apps. In order to analyze the reasons why some behaviors are classified as suspicious cases, we compare the behavior characteristics with common behaviors through data visualization and manual comparison, and obtain some results as follows.

**Finding1.** The 700 apps acquired in the AppStore that are generally considered benign, we detect 1,063 sensitive transmissions and 74.5% of these sensitive transmissions are related to advertising, as shown in Fig. 4. For example, a weather-forecasting app takes a user's location and sends it to an AD server.

**Finding2.** Some applications use specific environments to steal privacy. Such as an app from the DroidDream malware family only sends messages at night.

**Finding3.** Most malicious transcribes use resources that have clear semantics and are related to users' private information. To be specific, many malicious transcribes obtain the users' personal information and write them to files or

logs. Then malicious transcribes use APIs under NETWORK and SMS packets to transport sensitive information.

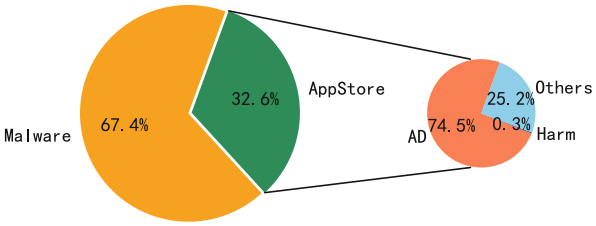



**Fig. 4.** A sensitive transport classification map of a dataset downloaded from AppStore.

## 5 Conclusion

This paper proposes TraceDroid, an Android malware detection system. At first we apply lightweight static analysis to get the entry points, and then perform dynamic analysis to track the traces of privacy leaks. To the best of our knowledge, our framework can identify anomalies in sensitive information instead of treating all sensitive information transmission as unreliable. We have conducted plenty of experiments to evaluate the performance of TraceDroid and results show that TraceDroid can effectively detect unknown malware samples with a 94% accuracy. However, there are still some problems to be explored, such as insufficient sample collection and the accuracy of online detection of unknown malware is low. In the next work, we will collect more samples, optimize our model, and improve the accuracy of unknown software detection.

## References

1. Cyber security statistics the ultimate list of stats, data & trends. purplesec.us/resources/cyber-security-statistics/
2. Arzt, S., et al.: Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. In: PLDI (2014)
3. Cai, Z., He, Z.: Trading private range counting over big iot data. In: 39th IEEE International Conference on Distributed Computing Systems, ICDCS 2019, Dallas, TX, USA, July 7–10, 2019. pp. 144–153. IEEE (2019). https://doi.org/10.1109/ICDCS.2019.00023
4. Cai, Z., He, Z., Guan, X., Li, Y.: Collective data-sanitization for preventing sensitive information inference attacks in social networks. IEEE Trans. Dependable Secur. Comput. **15**(4), 577–590 (2018). https://doi.org/10.1109/TDSC.2016.2613521
5. Cai, Z., Zheng, X.: A private and efficient mechanism for data uploading in smart cyber-physical systems. IEEE Trans. Netw. Sci. Eng. **7**(2), 766–775 (2020). https://doi.org/10.1109/TNSE.2018.2830307

6. Cai, Z., Zheng, X., Wang, J., He, Z.: Private data trading towards range counting queries in internet of things. IEEE Trans. Mob. Comput. (2022)
7. Chen, X., et al.: Android HIV: a study of repackaging malware for evading machine-learning detection. IEEE Trans. Inf. Forensics Security **15**, 987–1001 (2019)
8. Chen, X., Zhu, S.: Droidjust: automated functionality-aware privacy leakage analysis for android applications. In: WiSec (2015)
9. Enck, W., et al.: Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In: OSDI (2010)
10. Gordon, M.I., Kim, D., Perkins, J.H., Gilham, L., Nguyen, N., Rinard, M.C.: Information flow analysis of android applications in droidsafe. In: NDSS (2015)
11. Grace, M.C., Zhou, Y., Zhang, Q., Zou, S., Jiang, X.: Riskranker: scalable and accurate zero-day android malware detection. In: Davies, N., Seshan, S., Zhong, L. (eds.) The 10th International Conference on Mobile Systems, Applications, and Services, MobiSys'12, Ambleside, United Kingdom - June 25–29, 2012. pp. 281–294. ACM (2012). https://doi.org/10.1145/2307636.2307663
12. Kohavi, R., et al.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Ijcai (1995)
13. Lee, J., Lee, S., Lee, H.: Screening smartphone applications using malware family signatures. Comput. Secur. **52**, 234–249 (2015). https://doi.org/10.1016/j.cose.2015.02.003
14. Lu, K., et al.: Checking more and alerting less: Detecting privacy leakages via enhanced data-flow analysis and peer voting. In: NDSS (2015)
15. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Beyond blacklists: Learning to detect malicious web sites from suspicious urls. In: KDD (2009)
16. Mariconti, E., Onwuzurike, L., Andriotis, P., Cristofaro, E.D., Ross, G., Stringhini, G.: Mamadroid: Detecting android malware by building markov chains of behavioral models (2017)
17. Meng, Z., Xiong, Y., Huang, W., Qin, L., Jin, X., Yan, H.: Appscalpel: combining static analysis and outlier detection to identify and prune undesirable usage of sensitive data in android applications. Neurocomputing **341**, 10–25 (2019). https://doi.org/10.1016/j.neucom.2019.01.105
18. Pandita, R., Xiao, X., Yang, W., Enck, W., Xie, T.: WHYPER: towards automating risk assessment of mobile applications. In: King, S.T. (ed.) Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14–16, 2013. pp. 527–542. USENIX Association (2013). www.usenix.org/conference/usenixsecurity13/technical-sessions/presentation/pandita
19. Qu, Z., Rastogi, V., Zhang, X., Chen, Y., Zhu, T., Chen, Z.: Autocog: Measuring the description-to-permission fidelity in android applications. In: Ahn, G., Yung, M., Li, N. (eds.) Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3–7, 2014. pp. 1354–1365. ACM (2014). https://doi.org/10.1145/2660267.2660287
20. Raghuramu, A., Zang, H., Chuah, C.N.: Uncovering the footprints of malicious traffic in cellular data networks. In: PAM (2015)
21. Rasthofer, S., Arzt, S., Bodden, E.: A machine-learning approach for classifying and categorizing android sources and sinks. In: NDSS (2014)
22. Ren, J., Rao, A., Lindorfer, M., Legout, A., Choffnes, D.: Recon: Revealing and controlling pii leaks in mobile network traffic. In: MobiSys (2016)
23. Sihan, Q.: Research progress on android security. Ruan Jian Xue Bao J. Softw. **1**, 27 (2016)

24. Vallée-Rai, R. Co, P., Gagnon, E., Hendren, L.J., Lam, P., Sundaresan, V.: Soot - a java bytecode optimization framework. In: MacKay, S.A., Johnson, J.H. (eds.) Proceedings of the 1999 conference of the Centre for Advanced Studies on Collaborative Research, November 8–11, 1999, Mississauga, Ontario, Canada. p. 13. IBM (1999). dl.acm.org/citation.cfm?id=782008
25. Wang, Z., Li, C., Yuan, Z., Guan, Y., Xue, Y.: Droidchain: a novel android malware detection method based on behavior chains. Pervasive Mob. Comput. **32**, 3–14 (2016). https://doi.org/10.1016/j.pmcj.2016.06.018
26. Wüchner, T., Cislak, A., Ochoa, M., Pretschner, A.: Leveraging compression-based graph mining for behavior-based malware detection. IEEE Trans. Dependable Secur. Comput. **16**(1), 99–112 (2019)
27. Xia, M., Gong, L., Lyu, Y., Qi, Z., Liu, X.: Effective real-time android application auditing. In: S and P (2015)
28. Yang, W., Xiao, X., Andow, B., Li, S., Xie, T., Enck, W.: Appcontext: Differentiating malicious and benign mobile app behaviors using context. In: Bertolino, A., Canfora, G., Elbaum, S.G. (eds.) 37th IEEE/ACM International Conference on Software Engineering, ICSE 2015, Florence, Italy, May 16–24, 2015, Volume 1. pp. 303–313. IEEE Computer Society (2015). https://doi.org/10.1109/ICSE.2015.50
29. Zheng, X., Cai, Z.: Privacy-preserved data sharing towards multiple parties in industrial Iots. IEEE J. Sel. Areas Commun. **38**(5), 968–979 (2020). https://doi.org/10.1109/JSAC.2020.2980802
30. Zhou, Y., Jiang, X.: Dissecting android malware: Characterization and evolution. In: S and P (2012)
31. Zhou, Y., Wang, Z., Zhou, W., Jiang, X.: Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets. In: 19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, February 5–8, 2012. The Internet Society (2012), www.ndss-symposium.org/ndss2012/hey-you-get-my-market-detecting-malicious-apps-official-and-alternative-android-markets

# CA-Free Real-Time Fuzzy Digital Signature Scheme

Yijie Yan(✉) and Shiyuan Xu

School of Information Engineering, North China University of Technology, Beijing, China
yan020818@126.com

**Abstract.** Digital signature has been playing a very important role in electronic money transactions while memorizing and storing keys for digital signatures brings a great burden to users. In recent years, certifying authority (CA) has been attacked frequently, and many users have suffered losses. In this article, we discuss the existing digital signature technologies, analyze their disadvantages, and propose a novel fuzzy digital signature scheme, which has the characteristics of immediacy and no need for CA. Initially, our scheme uses fuzzy extractors to generate digital signatures seamlessly from biometric or other noise data, since its information entropy is sufficient to ensure security. Secondly, we reduce the dependence of digital signatures on third parties and the threats and delays caused by them. Furthermore, We also conducted the computational security analysis of the proposed scheme to illustrate its security level. Finally, the comprehensive experimental evaluation elaborates that our scheme is reliable and practical for real scenarios, ensuring the information measured by the user is untraceable.

**Keywords:** Digital signature · CA-free · Real-time signature · Fuzzy extractors · Applied cryptography · Information security

## 1 Introduction

With the advancement of the digital world and the introduction of the concept of meta-universe, security is still one of the important challenges we are facing. In increasingly digital transactions, we must ensure that such transactions are traceable and undeniable. Information authentication and privacy protection are important issues at present [15–17].

Signature schemes have been widely used not only in traditional communications but also in emerging fields such as the Internet of things and the blockchain [4, 14, 18, 19]. As an infrastructure for managing digital certificates, public key infrastructure (PKI) is essential to human beings. In this key system, the identity of the user depends on the private key to verify. The third-party certifying authority (CA) verifies and confirms the physical identity of the individual. After successful verification, CA issues a digital signature certificate (DSC), in which DSC appends a public key-private key pair signed by CA authentication. The security of cryptographic primitive depends on the security of the private key. Once the private key is compromised, the message will be exposed to


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 479–490, 2022.
https://doi.org/10.1007/978-3-031-19208-1_39

danger. Therefore, the user needs to protect the private key in a highly secure manner. For example, the user stores the private key in the USB token and remembers the token's personal identification number (PIN) to activate the key. Saving the key in this way will impose a burden on users, reduce availability, and be very unfriendly to the elderly.

Among the current mainstream digital signature models [1], two models are widely used: the first is a scheme based on offline encrypted tokens. The CA confirms the identity of the user by validating the physical document provided by the user and then distributes the DSC to store it on the encrypted token [10]. One of the main disadvantages of this scheme is that the verification process takes a long time, and more importantly, this method cannot prohibit unauthorized signature authorization, because the token can be transferred to anyone, regardless of whether that person is trusted or not. Therefore, the key cannot be securely saved in the hands of individuals or institutions that need DSC to be attached to the transaction. In addition, tokens can also be stolen, tampered with, or lost, and tokens need to be used in conjunction with other applications [2]. This scheme sets a period of validity for the token, and to ensure its security, users need to verify and update the certificate regularly, which makes the process very complicated. However, this scheme also has the advantage of the security of working offline. This solution does not require online external connections, and DSC has no external dependencies.

The second scheme is the online authentication scheme. CA authenticates to distribute the DSC through online authentication. The provider of this service is a reliable organization that has determined the identity of the user. The private key will be stored in a highly secure hardware security module (HSM) or discarded after the signature is completed. The weakness is very obvious, because the signature process is online, which requires the real-time availability of users and service providers, and may be subject to attacks such as deception, replay, and so on. There is another threat that must be considered in the above plan, and that is the risk from CA itself [3].

These disadvantages make it more important for us to find a secure scheme, which can work offline without external dependence, avoid the delay of offline authentication and prohibit unauthorized use. Biometric information is inherent and unique (such as fingerprint, iris, and finger vein), so using biometric data as a cryptographic private key is a promising method. Biometric information is not as easy to forget as traditional passwords and is more difficult to steal than USB tokens. Biometric features can be obtained through the application of sensors, and biometric templates (BT) can be generated by collecting these features [13]. BT provides the feature of generating digital signature keys.

For the problems mentioned above, our paper proposes a CA-free real-time fuzzy digital signature scheme. The contributions of this article are as listed follows.

- Taking biometric information or other fuzzy data as input, the signature process is completed and the sensitive data is protected from being stolen.
- The scheme proposed in this paper eliminates the requirements of the third-party pre-verification process. The signature process is separated from the external dependencies such as CA which are common in the current digital signature system.
- We have carried out security analysis and experiments on the scheme, and the results show that the scheme ensures the correctness and reliability of the signature. It is superior to the existing schemes in terms of cost, function, and process risk.

## 2 Preliminary

Some symbols used in the background are shown in Table 1.

**Table 1.** NOTATIONS

| Notation | Description |
|---|---|
| $\mathbb{N}$ | Natural number |
| $\mathbb{Z}$ | Integer number |
| $\mathbb{R}$ | Real number |
| $\phi$ | Feature space |
| $\varphi$ | Sample in feature space |
| $\max_\alpha \Pr[A = \alpha]$ | Predictability of random variable $A$ |
| $[i]$ | Set $\{1, 2, \cdots, i\}$ |
| $\varepsilon$ | Neglect |

**Definition 1 (Metric Space).** Metric space is a set $\phi$ with the following distance function:

$$\phi \times \phi \to \mathbb{R}^+$$

**Definition 2 (Min-Entropy).** [5] **.** For a random variable $A$, there is a minimum entropy:

$$H_\infty(A) = -\log(\max_\alpha \Pr[A = \alpha])$$

**Definition 3 (Average Min-Entropy).** [5]**.** In the case of the conditional distribution, the average minimum entropy of a given random variable $B, A$ is recorded as follows.

$$\underset{\infty}{\widetilde{H}}(A|B) = -\log(\underset{\beta \leftarrow B}{E}[2^{-H_\infty(A|B=\beta)}])$$

**Definition 4 (Statistical distance).** The statistical distance between the two probability distributions $A, B$ is:

$$SD(A, B) = \frac{1}{2}\sum |\Pr(A = \mu) - \Pr(B = \mu)|$$

### 2.1 Secure Sketches

The secure sketch is one of the components of the fuzzy extractor. The secure sketch scheme takes the noisy sample $\varphi$ as input and then outputs the sketch $s$, which can be used as a helping string. The purpose of the secure sketch is to restore $\varphi$ under s if and only if the input $\varphi'$ is close enough to $\varphi$.

**Definition 1 (Secure Sketches).** [8]**.** The secure sketch consists of $(SS, Rec)$ two-part algorithm.

$SS()$ takes $\varphi \in \phi$ as input and outputs a sketch $s \in \{0, 1\}^*$.

$Rec()$ takes $\varphi' \in \phi$ and $s$ obtained from $SS()$ as input and calculates $dis(\varphi, \varphi')$ if the result is less than the threshold $t$, then output $\varphi$ otherwise outputs -1.

**Definition 2 (Chebyshev distance).** Given two vectors $x = \{x_1, x_2, \cdots, x_n\}$ $y = \{y_1, y_2, \cdots, y_n\}$, the Chebyshev distance between them is defined as:

$$dis(x, y) = \max_i(|x_i - y_i|)$$

When $dis(x, y) \leq t; t \in \mathbb{R}^+$ it is said that $x$ and $y$ are close, where $t$ is the threshold.

### 2.2 Fuzzy Extractor Scheme

The secure sketch can reproduce the original input from a given noise data, and we can use this feature to construct a fuzzy extractor scheme [8], including $(Gen, Rep)$. It has the following properties:

The generating function $Gen()$ takes $\varphi \in \phi$ as input, output string $R \in \{0, 1\}^\eta$, and help string $P \in \{0, 1\}^*$.

The reproduction function $Rep()$ input $\varphi' \in \phi$ and help string $P \in \{0, 1\}^*$, output $R$, are:

$Rep(\varphi', P) \rightarrow R$ if $dis(\varphi, \varphi') \leq t$

**Table 2.** Composition function of the fuzzy extractor

**Algorithm 1** Composition function of the fuzzy extractor

1: $Gen()$ : **Input**: $\varphi \in \phi$
**Output**: $R \in \{0,1\}^\eta, P \in \{0,1\}^*$

2: $Rep()$ : **Input**: $\varphi' \in \phi$, $P \in \{0,1\}^*$
**If** $dis(\varphi, \varphi') \leq t$ :
**Output**: $R$

We review the general structure of the fuzzy extractor as shown in Table 3.

**Table 3.** The general structure of the fuzzy extractor

| **Algorithm 2** The general structure of the fuzzy extractor |
|---|
| 1: Setup: Let $SS$ be a secure sketch and $Ext$ be a strong extractor. |
| 2: $Gen()$ : **Input**: $\varphi \in \phi$ |
| **Output**: $Gen(\varphi; r_1, r_2) \to (P, R)$ |
| Among $P = (SS(\varphi; r_1), r2), R = Ext(x; r_2)$ |
| 3: $Rep()$ : **Input**: $\varphi' \in \phi$, $P \in \{0,1\}^*$ |
| Recover $\varphi = Rec(\varphi', SS(\varphi; r_1))$ |
| **Output**: $R = Ext(\varphi; r_2)$ |

## 3 Our Proposed Scheme

In this section, we will introduce our CA-free fuzzy signature scheme. The key pair generated by the RSA algorithm protects the authenticity and unforgeability of the message. Our proposed scheme has the following steps:

**Step 1.** The user uses the sensor to provide his biometrics (data with noise) and obtains $X$ times in a row.

**Step 2.** $X$ samples are converted into strings by the fuzzy extractor, the t of each sample is recorded, and the average number is calculated as the threshold.

**Step 3.** Generate a cancellable template (CT) using the converted string.

**Step 4.** Use CT to generate a public-private key pair.

**Step 5.** Calculate the hash of the file through the hash function, and encrypt it with the private key generated in the previous step to get the encrypted hash.

**Step 6.** Combine the encrypted hash, public key, encrypted CT, threshold, and master file obtained in the above steps to form the digitally signed document.

Figure 1. Shows the flow of our proposed scheme and the components of each link. Below we provide a detailed overview of the underlying methods of these steps.

The following describes the operation mode and principle of the key components in the scheme, as well as the signature verification method of the documents signed by the scheme.

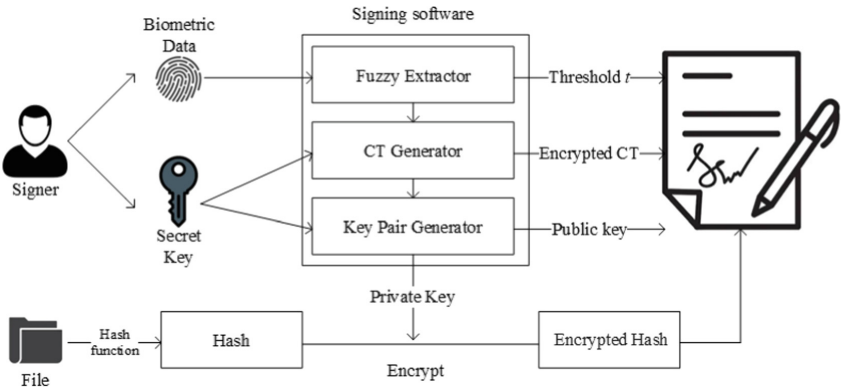



**Fig. 1.** The process of our proposed scheme

### 3.1 Fuzzy Extractor

The above conventional secure sketch scheme in part 2 can be transformed into a fuzzy extraction scheme, but there are security threats. That is, there is no security mechanism in the storage and transmission of data, and opponents can modify the data relatively easily [6]. Therefore, a more robust secure sketch scheme is introduced as shown in Table 4.

**Table 4.** Highly robust secure sketch

**Algorithm 3** Highly robust secure sketch

**Input:** $(SS, Rec)$

1: $setup()$ : Prepare a hash function $H: \{0,1\}^* \to \{0,1\}^\sigma$.

2: $ss'(\varphi) \to s'$ include:

$SS(\varphi) \to s$

$h = H(\varphi, s)$

$s' = (s, h)$

3: $Rec'(\chi, s') \to \varphi$ include:

$Rec(\chi, s) \to \varphi'$

**if** $\varphi' = -1$ **return** -1

**else: return** $\varphi = \varphi'$

Through this new secure sketch scheme, we can get a more secure fuzzy extractor scheme as shown in Table 5.

**Table 5.** Fuzzy extractor scheme with high robustness

**Algorithm** 4 Fuzzy extractor scheme with high robustness

1: Setup: Let $Ext$ be a strong extractor. It runs the same system setup as the robust secure sketch above.

2: $Gen()$ : Select a random string $str \in \{0,1\}^{\eta}$ of $\eta$ bits from a given sample $\varphi$ .

Using $SS'()$ get $s'$

calculate $R = Ext(\varphi, str)$ $P = (s', str)$

**return** $(R, P)$

3: $Rep()$ : Input: $\chi \in \phi$ , $P$

Using $Rec'()$ get $\alpha$

reproduces the string $R$ by computing $E(\alpha, str)$

### 3.2 CT Generator

After the $X$ samples were extracted by the fuzzy extractor, $X(R, P)$ were obtained.

$$CT = \sum_{i=1}^{X} F_{\alpha}((R_i, P_i), secret\_key)$$

The $X$ samples are grouped together and out of order to generate $CT$. The function $F_{\alpha}$ has the identity, and the same $CT$ can always be generated for the combination of a given key and $(R, P)$. For different keys, even if the input samples are the same, they will not get the same $CT$.

### 3.3 Key Pair Generator

The public-private key pair generated by the key pair generator contains two 1024-bit strong primes $p, q$. It is obtained by using function $F_{\beta}$.

$$(p, q) = F_{\beta}(CT, secret\_key)$$

$F_{\beta}$ is an irreversible function, and the prime pairs calculated are not close to each other [11, 12]. The key pair generator then uses these two primes to generate the public key and private key of the destination length through the traditional RSA algorithm.

### 3.4 Signed File Verification

When the correctness of the file is in doubt, we need to verify the file to know whether it has been tampered with or not. If the signatory has countersigned or stored a signature sample in another verifying authority or a trusted third party, then the signature comparison will be carried out by the agency.

The rest of the cases are verified using the properties described above:

*Verify whether CT is valid.* The signer re-provided its fuzzy sample and key again on the spot, reuse the fuzzy extractor, and verify whether it is below the threshold. If it falls below the threshold, it is considered a match, meaning that the signer is the person who signed this document.

*Verify the public-private key pair.* The public-private key pair is generated from the secret_key owned by the signer and stored $CT$. The generated public key is then paired with the private key previously used for signature, and the public key is used to decrypt the hash. When the decrypted hash values match, the verification is successful.

## 4 Security Analysis

In this section, we analyze the popular vulnerabilities that the proposed model may face. The transmission of encryption keys in the public channel makes them vulnerable to man-in-the-middle attacks, and the generation and use of encryption keys locally by signers can avoid such threats. In addition, we use the $CT$ derived from the sample to generate the encryption key, which can avoid storing the sample itself in external devices and eliminate the theft and modification of the mandarin information contained in the sample, or even other more serious threats.

We assume that an opponent obtains the secure sketch in some way and tries to restore the sample features of its input.

**Definition 1.** For a secure sketch $(\phi, \varphi, \widetilde{\varphi}, t)$. As far as it is concerned, if the arbitrarily distributed $W$ on the metric space $\phi$ has a minimal entropy $\varphi$ and the opponent has the advantage of $2^{-\widetilde{m}}$ at most, $W$ can be reduced by $\widetilde{m} \leq \widetilde{H}_{\infty}(W|SS(W))$

The security of fuzzy extractors is based on statistical indistinguishability.

**Definition 2.** The fuzzy extractor $(\phi, \varphi, \eta, t)$ is safe. Enter any distribution $W$ on the metric space $\phi$ with minimum entropy $\varphi$, the output string is in the distribution $R$, and the statistical distance between $R$ and the uniform distribution $U_\eta$ is negligible.

$$SD((R, P), (U_\eta, P)) \leq \varepsilon$$

**Theorem 1.** If the secure sketch scheme is secure, then the fuzzy extractor scheme is also secure.

***Proof*:** We use the secure sketch proposed by the general construction [8] and the fuzzy extractor scheme derived. This general structure ensures the security of the scheme, that is, the output string of the scheme is indistinguishable from the randomly generated string.

$CT$ is not used directly as an encryption key. It is used to derive two prime numbers needed to generate an asymmetric encryption RSA algorithm [9]. The security of the key is based on the security of the RSA algorithm. The user-specified key is used to export two 1024-bit primes from $CT$. The resulting prime number is about 150 digits long.

**Definition 3.** It is less likely that two users will generate similar keys through similar primes. We can estimate the probability of conflict by calculating the prime density, using $\frac{1}{\ln(n)}$. There are $\frac{10^{150}}{\ln(10^{150})}$ primes, about $2.7 \times 10^{147}$. Therefore, the probability of two users getting the same prime pair is very small.

**Theorem 2.** The proposed fuzzy digital signature scheme is secure, if the underlying security sketch scheme, fuzzy extractor scheme, and key generation scheme are secure.

***Proof*:** If the key generator used is secure, then the *CT* as input is unrecoverable because the proposed fuzzy extractor scheme is secure. In the whole process of the scheme, the identification device runs the $SS()$ algorithm to get the sketch $s$ of the user, because the proposed security sketch is proved to be secure, the attacker cannot disclose the user's biometric information. Therefore, the proposed fuzzy digital signature scheme is secure.

## 5 Experiment Results

The database used in the experiment is from FVC2002. FVC2002 is the Second fingerprint verification competition [7]. In this study, only FVC2002DB1, DB2, DB3, and DB4 were used for the experiment. The database contains eight impressions for each finger of different people, as shown in the following Table 6.

**Table 6.** Scanners/technologies used for the collection of databases

| | Technology | Scanner | Image Size – Res |
|---|---|---|---|
| DB1 | Optical | Identix TouchView | 388 × 374 - 500dpi |
| DB2 | Optical | Biometrika FX2000 | 296 × 374 - 569dpi |
| DB3 | Capacitive | Precise Biometrics | 300 × 300 - 500dpi |
| DB4 | Synthetic | SFinGE | 288 × 384 - 500dpi |

We have done signature experiments on the samples from four databases, recorded the characteristic data of *CT* generated by them, compared them, and got Table 7.

**Table 7.** Test results from the database

| Database name | Average entropy of CT | Time consumption | Average length of CT |
|---|---|---|---|
| DB1 | .9756 | .002 | 5495 |
| DB2 | .9446 | .002 | 8456 |
| DB3 | .9632 | .001 | 5159 |
| DB4 | .9057 | .003 | 4741 |

In addition, we randomly select a finger in the database and collect its different impressions. Through Fig. 2, we can see that Experiment 6–10 are always below the threshold while using mismatched data, such as other fingers, such as Experiment 1–5 is always higher than the threshold.

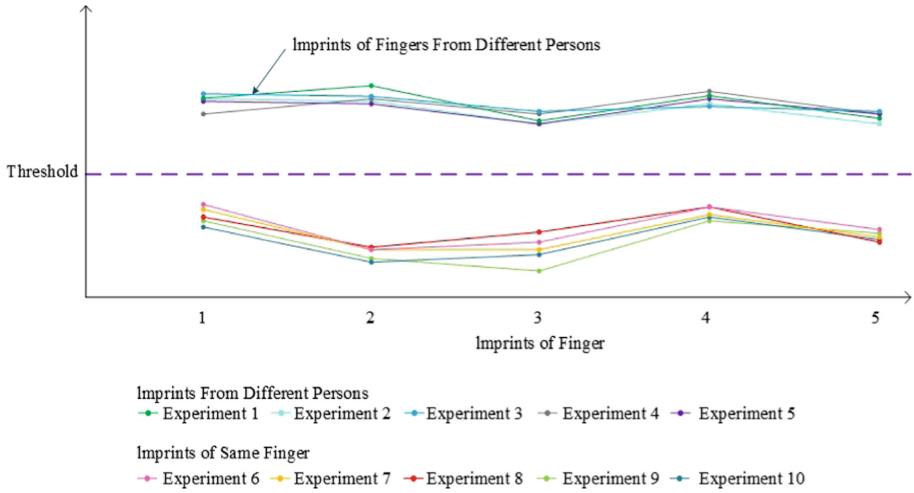



**Fig. 2.** Comparison between different samples and threshold

## 6 Conclusion

In this paper, we discussed the disadvantages of the current mainstream digital signature schemes, and then we design a CA-free real-time fuzzy digital signature scheme. We avoid the intervention of the third party in the signature process, get rid of the external dependence, and real-time signature reduces the time consumption of the digital signature process. The security of the scheme depends on the strength of the RSA algorithm and the security of the secure sketch and fuzzy extractor schemes. Finally, we carried out verification experiments and security analysis of the scheme. The results show that our scheme achieves the design goal, ensuring that the user's signature is reliable, and has superior characteristics compared with other existing protocols.

## References


1. Arroyo, D., Diaz, J., Rodriguez, F. B.: Non-conventional digital signatures and their implementations—a review. In: Computational Intelligence in Security for Information Systems Conference, pp. 425–435. Springer, Cham, Switzerland (2015). https://doi.org/10.1007/978-3-319-19713-5_36
2. Bakshi, P., Subramanian, N., Nandi, S.: Using digital tokens to improve amortized performance of eSign. In: 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), pp. 121–128. IEEE, Athens, Greece (2018)
3. Saadatm, An, J., Rahimi, A.: Digital certificate of public key for user authentication and session key establishment for secure network communications. Int. J. Netw. Security **23**(3), 480–489 (2021)
4. Xu, S., Chen, X., Wang, C., He, Y., Xiao, K., Cao, Y.: A lattice-based ring signature scheme to secure automated valet parking. In: Liu, Z., Wu, F., Das, S.K. (eds.) Wireless Algorithms, Systems, and Applications. Lecture Notes in Computer Science, vol. 12938, pp. 70–83. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86130-8_6
5. Shoup, V.: A Computational Introduction to Number Theory and Algebra. Cambridge University Press, Cambridge (2009)
6. Takahashi, K., Matsuda, T., Murakami, T., Hanaoka, G., Nishigaki, M.: Signature schemes with a fuzzy private key. Int. J. Inf. Secur. **18**(5), 581–617 (2019). https://doi.org/10.1007/s10207-019-00428-z
7. Maio, D., Maltoni, D., Cappelli, R., Wayman, J. L., Jain, A. K.: FVC2002: Second fingerprint verification competition. In: Object recognition supported by user interaction for service robots, pp. 811–814. IEEE, Quebec City, Canada (2002)
8. Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., Smith, A.: Secure remote authentication using biometric data. In: Cramer, R. (ed.) Advances in Cryptology – EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, pp. 147–163. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_9
9. Boneh, D.: Twenty years of attacks on the RSA cryptosystem. Notices of the AMS **46**(2), 203–213 (1999)
10. Khan, A.A.: Preventing phishing attacks using one time password and user machine identification. Int. J. Comput. Appl. **68**(3), 7–11 (2013)
11. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J.L. (eds.) Advances in Cryptology - EUROCRYPT 2004. Lecture Notes in Computer Science, vol. 3027, pp. 523–540. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_31
12. Sarkar, A., Singh, B.K.: Cryptographic key generation from cancelable fingerprint templates. In: 2018 4th International Conference on Recent Advances in Information Technology (RAIT), pp. 1–6. IEEE, Dhanbad, India (2018)
13. Terhörst, P., Fährmann, D., Damer, N., Kirchbuchner, F., Kuijper, A.: Beyond identity: What information is stored in biometric face templates? In: 2020 IEEE International Joint Conference on Biometrics (IJCB), pp. 1–10. IEEE, Houston, TX (2020)
14. Xu, S., Chen, X., He, Y.: EVchain: an anonymous blockchain-based system for Charg-ing-Connected electric vehicles. Tsinghua Sci. Technol. **26**(6), 845–856 (2021)
15. Cao, Y., Xu, S., Chen, X., He, Y., Jiang, S.: A forward-secure and efficient authentication protocol through lattice-based group signature in VANETs scenarios. Comput. Netw. **124** (2022)

16. Cheng, Y., Xu, S., Zang, M., Jiang, S., Zhang, Y.: Secure Authentication Scheme for VANET Based on Blockchain. In: Proceedings of ICCC, pp. 1526–1531 (2021)
17. Cheng, Y., Xu, S., Zang, M., Kong, W.: LPPA: A Lightweight Privacy-Preserving Authentication Scheme for the Internet of Drones. In: Proceedings of ICCT, pp. 656–661 (2021)
18. Chen, X., Xu, S., Qin, T., Cui, Y., Gao, S., Kong, W.: AQ-ABS: Anti-Quantum Attribute-based Signature for EMRs Sharing with Blockchain. In: Proceedings of WCNC, pp. 1176–1181 (2022)
19. Chen, X., Xu, S., He, Y., Cui, Y., He, J., Gao, S.: LFS-AS: Lightweight Forward Secure Aggregate Signature for e-Health Scenarios. In: Proceedings of ICC, early access (2022)

# Efficient Post Quantum Random Oblivious Transfer Based on Lattice

Lidong Xu and Mingqiang Wang(✉)

School of Mathematics, Shandong University, Jinan 250100, Shandong, China
xulidong@mail.sdu.edu.cn, wangmingqiang@sdu.edu.cn

**Abstract.** The large scale multiparty computation and private set intersection requires a number of oblivious transfer instances as subroutines, but the implementation of oblivious transfer protocols is relatively slow. An feasible way is to use the oblivious transfer variant called random oblivious transfer. In this paper, we propose a 1-out-of-2 random oblivious transfer protocol and extend it to a 1-out-of-$k$ random oblivious transfer protocol based on the LWE assumption, quantum computation and measurement. Then, we analysis the stand-alone security of our 1-out-of-2 random oblivious transfer protocol under various malicious situations and prove its universally composable security in UC framework. As for the security of our 1-out-of-$k$ random oblivious transfer protocol, the similar results can be obtained.

**Keywords:** Oblivious transfer · LWE problem · Quantum computation · UC-security

## 1 Introduction

Oblivious transfer (OT) is an important cryptographic primitive which can be used for designing secure multi-party computation (MPC) [1–3], bit commitment [4–6] and private set intersection (PSI) [7,8]. The OT protocol was firstly proposed, by Michael O. Rabin in 1981, to construct a secrets exchange scheme [9]. The original OT protocol has two participants, where one party (the sender) sends a message to another (the receiver) with the requirement that the receiver obtains this message with probability $\frac{1}{2}$ and the sender remains oblivious of whether the message has been received or not.

In order to build protocols for secure two-party computation, a more useful kind of OT protocol, called the 1-out-of-2 OT protocol, was developed [10–13]. In these protocols, the receiver is allowed to get one message from the sender's message pair without knowing anything about the other message, and the sender

Supported by the National Key Research and Development Program of China (No. 2021YFA1000600), the National Key Research and Development Program of China (Grant No. 2018YFA0704702), and the National Natural Science Foundation of China (Grant No. 61832012).


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 491–499, 2022.
https://doi.org/10.1007/978-3-031-19208-1_40

is required not to know about the receiver's choice. Another OT variant is the randomized oblivious transfer (ROT), the only difference from 1-out-of-2 OT lies in that the receiver is required to get one message randomly.

As is know, MPC protocols based on oblivious-circuit evaluation techniques require a large number of OT. Since the OT schemes are relatively slow, they become a major bottleneck for the large-scale MPC implementations. In order to deal with the problem of OT efficiency, Ishai et al. introduce the concept of OT extension [14] where one needs to use ROT instances as base OTs. In addition, the ROT scheme also is a main tool in designing efficient PSI protocols [8] which is one of the most popular MPC technique.

Motivated by the construction of trapdoor, claw free, 2-regular functions in [15–17], we propose a 1-out-of-2 ROT protocol based on quantum mechanics and LWE assumption. Then, we construct a family of trapdoor claw-free $k$-regular functions and extend the 1-out-of-2 ROT protocol to a 1-out-of-$k$ ROT protocol. Furthermore, we analysis the stand-alone security of our ROT protocols under various malicious situations and prove their universally composable security in UC framework. The key technique of our protocol is to construct a family of trapdoor, claw free, $k$-regular function based on the LWE assumption. Another technique used in our protocol is quantum computation and quantum entanglement by which Bob can obtain only one of $k$ preimages after measuring the produced quantum state.

## 2 The Construction of TCF $k$-Regular Functions

In this section, we will describe the construction of trapdoor claw-free (TCF) 2-regular functions defined in [17] and the construction of trapdoor claw-free $k$-regular functions, which are necessary for our ROT protocols. We start with the definition of trapdoor claw-free $k$-regular functions as follows:

**Definition 1** *(Trapdoor claw-free $k$-regular). A deterministic function $f : D \to R$ is a trapdoor claw-free $k$-regular function if the following conditions hold:*

- *$k$-regular: $\forall y \in Im(f)$, we have $|f^{-1}(y)| = k$.*
- *collision resistance: It is impossible to find out any pair $(x_0, x_1)$ such that $x_0 \neq x_1 \wedge f(x_0) = f(x_1)$ for any QPT algorithm without the trapdoor.*
- *Trapdoor one-way: Given $y \in Im(f)$ and the trapdoor $t_f$ of the function $f$, there exists a QPT algorithm that can return the set $f^{-1}(y)$. Moreover, it is impossible to get any $x \in f^{-1}(y)$ for any QPT algorithm without the trapdoor.*

### 2.1 Requirements on Parameters

Let $\lambda \in \mathbb{Z}$ be the security parameter in the LWE problem, all other parameters be the functions of $\lambda$.

- $n = \lambda$, the length of vector $\mathbf{s}$ ;
- $q = poly(n)$, the prime modulus;

- $m \approx 2n \lg q$, the length of the error vector $\mathbf{e}$;
- $\alpha \in (0,1)$, the discrete Gaussian distribution $\overline{\Phi}_\alpha$ is centered around 0 with standard deviation $\alpha q \geq 2\sqrt{n}$.

Under the setting of the parameters above, the LWE problem is at least as hard as solving SIVP [18,19]. And thus, the functions constructed in Sect. 3.2 and Sect. 3.3 are all trapdoor claw-free.

### 2.2 On the TCF 2-Regular Functions

In [17], the authors constructed a family $\mathcal{F}_2$ of TCF 2-regular functions based on the existence of a family $\mathcal{G}$ of injective, homomorphic, trapdoor one-way functions. For the completeness, we will recall the construction of $\mathcal{F}_2$ and related knowledge in this subsection.

The specific family $\mathcal{G}$ of injective, homomorphic, trapdoor one-way functions was constructed by Micciancio and Peikert [20]. Here, we list the outline of their construction and leave the detail to readers. First, generate a $n \times \bar{m}$ matrix $A$ by randomly choosing its elements from $\mathbb{Z}_q$ and a $\bar{m} \times kn$ trapdoor matrix $R$ by sampling its elements from a discrete Gaussian distribution $\mathcal{D}_{\alpha q}^{\bar{m} \times \omega}$ with mean 0 and standard deviation $\alpha q$. Then, select a fixed matrix $G$ as in [20] for which the function $g_G(s,e) = s^t G + e^t$ can be efficiently inverted, and construct the index matrix $K$ by concatenating $A$ and $G - AR$, i.e. $K = (A, G - AR)$. Finally, define the function $g_K$ with trapdoor $t_K = R$, which forms the family $\mathcal{G}$, as follow:

$$g_K(s,e) = s^t K - e^t, \tag{1}$$

where $s \in \mathbb{Z}_q^n$ and $e \in L^m$, $L$ is the domain of the errors in the LWE problem (the set of integers bounded in absolute value by $\mu$).

**Theorem 1 (*[20]*).** *The functions in $\mathcal{G}$ are injective, homomorphic, trapdoor one-way.*

### 2.3 The Construction of TCF $k$-Regular Functions

In order to design the $1-k$ ROT protocol, we need to construct a family $\mathcal{F}_k$ of TCF $k$-regular functions. Motivated by the idea of constructing TCF 2-regular functions in Sect. 3.2, we construct the family $\mathcal{F}_k$ also by using the family $\mathcal{G}$ of homomorphic injective trapdoor one-way functions.

Let $g_K \in \mathcal{G}$ with trapdoor $t_K$, $x^i \in \mathcal{D} \setminus \{0\} (0 \leq i \leq k-2)$ satisfying $x^i \neq x^j$ whenever $i \neq j$, we define the function $f : \mathcal{D} \times \mathbb{Z}_k \to \mathcal{R}$ with trapdoor $t_f = (t_K, x^0, ..., x^{k-1})$, which forms the family $\mathcal{F}_k$, as follows:

$$f(x,c) = \begin{cases} g_K(x), & \text{if c = 0;} \\ g_K(x) + g_K(x^0), & \text{if c = 1;} \\ g_K(x) + g_K(x^1), & \text{if c = 2;} \\ ... & \\ g_K(x) + g_K(x^{k-2}), & \text{if c = k-1.} \end{cases} \tag{2}$$

In a similar way as proving the functions in $\mathcal{F}_2$ are TCF 2-regular in [17], we can prove that the functions in $\mathcal{F}_k$ constructed above is TCF $k$-regular.

**Theorem 2.** *The functions in the family $\mathcal{F}_k$ are trapdoor claw-free $k$-regular.*

## 3 Our $1-k$ ROT Protocols

In this section, we will present a $1-2$ ROT protocol by using the family $\mathcal{F}_2$ of TCF 2-regular functions in [17], and extend this protocol into a $1-k$ ROT protocol by using the family $\mathcal{F}_k$ of TCF $k$-regular functions constructed in Sect. 3. As in [17], for $k \geq 2$, we employ the family $\mathcal{F}_k$ of TCF $k$-regular functions in a convenient form as $\mathcal{F}_k = \{f : \{0,1\}^n \to \{0,1\}^m\}$, where the domain of each $f$ is also denoted by $D$.

### 3.1 The $1-2$ ROT Protocol

In the prepare stage, first choosing a fixed function $f$ and its trapdoor $t_f$ from the family $\mathcal{F}_2 = \{f : \{0,1\}^n \to \{0,1\}^m\}$ of TCF 2-regular functions. Then, giving $(f, t_f)$ to the sender Alice and $f$ to the receiver Bob. To transfer the two messages $b_1, b_2 \in \{0,1\}^m$ from Alice to Bob obliviously, our $1-2$ ROT protocol performs as follows:

1. Bob prepares his registers at $\frac{1}{\sqrt{|D|}} \sum_{x \in D} (|x\rangle \otimes |0\rangle)$.
2. Bob applies the operator $U_f$ by using the first register as control and the second one as target, and the state in the two registers is in the form of $\frac{1}{\sqrt{|D|}} \sum_{x \in D} |x\rangle|f(x)\rangle$. After that, Bob sends the second register to Alice.
3. Alice measures her register in the computational basis and obtains the outcome $y$. Then, Bob's register becomes $\frac{1}{\sqrt{2}}(|x_1\rangle + |x_2\rangle)$, where $f(x_1) = f(x_2) = y$. Bob measures his register in the computational basis and obtains the outcome $\widetilde{x}$ ($= x_1$ or $x_2$).
4. Alice computes the preimages $x_1, x_2$ of $y$ by using the trapdoor $t_f$. Then, she sends the pairs $(a_1 = b_1 \oplus x_1, h(x_1))$ and $(a_2 = b_2 \oplus x_2, h(x_2))$ to Bob, where $h(x)$ represents the last bit of $x$.
5. Bob computes the value of $f(a_1 \oplus a_2 \oplus \widetilde{x})$. If the result is $y$ (which means $b_1 = b_2$), then he terminates this protocol.
6. Bob gets the message $b_\sigma$ by computing $a_\sigma \oplus \widetilde{x}$ if $h(x_\sigma) = h(\widetilde{x})$ ($\sigma = 1$ or 2).

### 3.2 The $1-k$ ROT Protocol

To extend the protocol above into the general $1-k$ ROT protocol, we only need to substitute the TCF 2-regular function for a TCF $k$-regular function constructed in Sect. 3.3.

In the prepare stage, first choosing a fixed function $f$ and its trapdoor $t_f$ from the family $\mathcal{F}_k = \{f : \{0,1\}^n \to \{0,1\}^m\}$ of TCF $k$-regular functions. Then, giving $(f, t_f)$ to the sender Alice and $f$ to the receiver Bob. To transfer the $k$ messages $b_1, b_2, ..., b_k \in \{0,1\}^m$ from Alice to Bob obliviously, our $1-k$ O.T. protocol performs as follows:

1. Bob prepares his registers at $\frac{1}{\sqrt{|D|}}\sum_{x\in D}(|x\rangle\otimes|0\rangle)$.
2. Bob applies the operator $U_f$ by using the first register as control and the second one as target, and the state in the two registers is in the form of $\frac{1}{\sqrt{|D|}}\sum_{x\in D}|x\rangle|f(x)\rangle$. After that, Bob sends the second register to Alice.
3. Alice measures her register in the computational basis and obtains the outcome $y$. Then, Bob's register becomes $\frac{1}{\sqrt{k}}(|x_1\rangle+...+|x_k\rangle)$ where $f(x_1)=...=f(x_k)=y$. Bob measures his register in the computational basis and obtains the outcome $\widetilde{x}(\in\{x_1,x_2,...,x_k\})$.
4. Alice computes the preimages $x_1,...,x_k$ of $y$ by using the trapdoor $t_f$. Then, she sends the pairs $(a_i=b_i\oplus x_i,h(x_i))(1\le i\le k)$ to Bob, where $h(x)$ presents the last $\lfloor\log k\rfloor$ bits of $x$.
5. Bob computes the value of $f(a_i\oplus a_j\oplus\widetilde{x})(1\le i<j\le k)$. If some $f(a_i\oplus a_j\oplus\widetilde{x})=y$ (which means $b_i=b_j$), then he terminates this protocol.
6. Bob gets the message $b_\sigma$ by computing $a_\sigma\oplus\widetilde{x}$ if $h(x_\sigma)=h(\widetilde{x})$ where $\sigma\in\{1,2,...,k\}$.

### 3.3 The Security Analysis of Our $1-2$ ROT Protocol

In this section, we will consider the stand-alone security of our $1-2$ ROT protocol in two aspects, Bob's malicious operation and Alice's malicious operation. The extended version, $1-k$ ROT protocol, can be analysed in the same way. Let us first recall the following property of the family $\mathcal{F}_2$ described in Sect. 3.2, on which the security of our $1-2$ ROT protocol is based.

**Theorem 3** *[17]. The functions in the family $\mathcal{F}_2$ described in Sect. 3.2 are TCF 2-regular.*

**Bob's Malicious Strategy.** A malicious receiver Bob aims to get both two messages $b_1$ and $b_2$ from Alice. To achieve his aim, Bob has to find a method to get the collision $x'$ for his measurement outcome $\widetilde{x}$ in Step 3. Except for guessing $x'$, what he could do is computing $y=f(\widetilde{x})$, and managing to find the preimages of $y$ with respect to $f$. But, the function $f$ is one-way according to Theorem 3, and thus Bob cannot obtain the preimages of $y$ by inverting $f$. So, it is impossible that Bob have an efficient method to get both $b_1$ and $b_2$.

**Alice's Malicious Strategy.** A malicious sender Alice wants to know what message Bob gets from the transfer procedure. There are two ways for Alice to achieve her aim, one is to get Bob's measurement outcome $\widetilde{x}$ and another is to cheat by sending illegal information to Bob in Step 4.
Note that, Alice gets $y$ by measuring her register and Bob obtains $\widetilde{x}$ by measuring his register with the superposition state $\frac{1}{\sqrt{2}}(|x_1\rangle+|x_2\rangle)$ in Step 3. Although Alice can computes the preimages $x_1$ and $x_2$ of $y$ by the trapdoor $t_f$ in Step 5, and $\widetilde{x}$ must be one of $x_1$ and $x_2$, Alice has no way to determine which one $\widetilde{x}$ is. So, the first way is not possible.

As for the second way, Alice may send two pairs $(a_1 = b_1 \oplus w_1, h(w_1))$ and $(a_2 = b_2 \oplus x_2, h(x_2))$ with $b_1 = b_2$ to Bob in Step 4. If Bob does not verify whether the two pairs are legal, he will always get $b_1$ in Step 6, no matter what his measurement outcome $\widetilde{x}$ is. And thus, Alice can know what the message Bob obtains. But in Step 5, Bob verifies the reality of the two pairs from Alice by computing the value of $f(a_1 \oplus a_2 \oplus \widetilde{x})$. If the result is $y$, then Bob infers that $b_1$ and $b_2$ are the same, and terminates the protocol. Therefore, this strategy also does not work.

## 4 The UC-security of Our $1-2$ ROT Protocol

In this section, we will prove the universally composable security of our $1-2$ ROT protocol in the UC framework. As for our $1-k$ ROT protocol, its UC-security can be proven in the same way.

We work in the standard universal composability framework of Canetti [21] with static corruption of some parties. The ideal world execution involves dummy parties (some of whom may be corrupted by an ideal adversary) interacting with the functionality $\mathcal{F}$. The dummy parties only relay the inputs to $\mathcal{F}$, and relay the outputs of $\mathcal{F}$ to the calling machine. The real world execution involves parties (some of whom may be corrupted by a real world adversary) interacting only with each other.

For our $1-2$ ROT protocol interacting with an adversary, the functionality $\mathcal{F}_{ROT}$ interacting with the simulator in the ideal world is defined as follows:

**Functionality $\mathcal{F}_{ROT}$**

**Parameters:** String length $n$.
**Parties:** The sender Alice and the receiver Bob.

1. Upon receiving the message $b_0, b_1$ from Alice and activated by Bob, $\mathcal{F}_{ROT}$ outputs $b_\sigma$ to Bob randomly

**Fig. 1.** The functionality $\mathcal{F}_{ROT}$

Let $\mathcal{A}$ be a static adversary that interacts with the parties Alice and Bob running the $1-2$ ROT protocol, we now construct a simulator $\mathcal{S}$ in ideal world interacting with the ideal functionality $\mathcal{F}_{ROT}$, such that no environment $\mathcal{Z}$ can distinguish the interaction with $\mathcal{A}$ in the real world from the interaction with $\mathcal{S}$ in the ideal world. The simulator $\mathcal{S}$ starts by invoking a copy of $\mathcal{A}$ and runs a simulated interaction of $\mathcal{A}$ with $\mathcal{Z}$ and the parties Alice and Bob. More specifically, the simulator $\mathcal{S}$ works as follows:

**Simulating the communication with $\mathcal{Z}$:** Every input value that $\mathcal{S}$ receives from $\mathcal{Z}$ is written on the adversary $\mathcal{A}$'s input tape (as if coming from $\mathcal{A}$'s environment). Every output value written by $\mathcal{A}$ on its output tape is copied to $\mathcal{S}$'s output tape (to be read by the environment $\mathcal{Z}$).

**Simulating the case when only Alice is corrupted:** The simulator $\mathcal{S}$ randomly selects a function $f$ with its trapdoor $t_f$ from the family $\mathcal{F}_2$ of TCF 2-regular functions, and sends $(f, t_f)$ to Alice and $f$ to Bob respectively.
When $\mathcal{A}$ produces $(a_1, w_1)$ and $(a_2, w_2)$ with $w_1 \neq w_2$ for honest Bob, $\mathcal{S}$ randomly chooses some $\widetilde{x} \in D$. Then, $\mathcal{S}$ computes $y = f(\widetilde{x})$ and another preimage $\widetilde{x'}$ of $y$ by the trapdoor $t_f$. After that, $\mathcal{S}$ computes $b_1 = a_1 \oplus \widetilde{x}$ and $b_2 = a_2 \oplus \widetilde{x'}$ where $h(w_1) = h(\widetilde{x}), h(w_2) = h(\widetilde{x'})$ and stores them. When dummy Bob is activated, $\mathcal{S}$ sends $b_1$ and $b_2$ to $\mathcal{F}_{ROT}$. When $\mathcal{F}_{ROT}$ returns $b_\sigma$, $\mathcal{S}$ outputs it as if from Bob.
**Simulating the case when only Bob is corrupted:** The simulator $\mathcal{S}$ randomly selects a function $f$ and its trapdoor $t_k$ from the family $\mathcal{F}_2$ of TCF 2-regular functions, and sends $(f, t_f)$ to Alice and $f$ to Bob respectively.
When the dummy Alice is activated, $\mathcal{S}$ gets $b_\sigma$ from the functionality $\mathcal{F}_{ROT}$ and stores it. When $\mathcal{A}$ is activated, $\mathcal{S}$ outputs $b_\sigma$ as if from Bob.
**Simulating the remaining cases:** When both parties are corrupted, the simulator just runs $\mathcal{A}$ internally (who itself generates the messages from both Alice and Bob). When neither party is corrupted, there is no necessity to construct $\mathcal{S}$. According to the above models of different corrupted cases, we obtain the following two propositions. And thus, our $1-2$ ROT protocol possesses the UC-security.

**Proposition 1.** *If the adversary $\mathcal{A}$ corrupts Alice in an execution of our $1-2$ ROT protocol $\pi$, then we have*

$$\boldsymbol{IDEAL}_{\mathcal{F}_{ROT},\mathcal{S},\mathcal{Z}} \overset{s}{\approx} \boldsymbol{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}.$$

**Proposition 2.** *If the adversary $\mathcal{A}$ corrupts Bob in an execution of our $1-2$ ROT protocol $\pi$, then we have*

$$\boldsymbol{IDEAL}_{\mathcal{F}_{ROT},\mathcal{S},\mathcal{Z}} \overset{s}{\approx} \boldsymbol{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}.$$

**Theorem 4.** *Denote our $1-2$ ROT protocol as $\pi$, then*

$$\boldsymbol{IDEAL}_{\mathcal{F}_{ROT},\mathcal{S},\mathcal{Z}} \overset{s}{\approx} \boldsymbol{EXEC}_{\pi,\mathcal{A},\mathcal{Z}}.$$

*Thus, $\pi$ UC-emulates the ideal function $\mathcal{F}_{ROT}$, in other word, $\pi$ is UC-secure.*

## 5 Conclusion

Motivated by the construction of trapdoor claw-free 2-regular functions in [17], we propose a $1-2$ ROT protocol and construct a family of trapdoor claw-free $k$-regular functions based on which we extend the $1-2$ ROT protocol to the $1-k$ ROT protocol. In our protocols, the key techniques are quantum computation and the family of trapdoor, claw free, $k$-regular functions. Furthermore, We analysis the stand-alone security of our $1-2$ ROT protocol in various malicious situations and prove its composable security in the UC framework. Certainly, the security of our $1-k$ ROT protocol can be obtained by a similar discussion.

Comparing with other OT protocols, our $1-2$ ROT protocol possesses stronger security and needs fewer rounds between the sender and the receiver. We give an intuitional comparison between our $1-2$ ROT protocol and the others presented before in the following table:

**Table 1.** Comparison with other OT (ROT) protocols

| Protocol | Round (moves) | Security |
|---|---|---|
| OT in [22] | 5 (including 2 with functionality) | UC-secure |
| OT in [23] | 6 | Non proof |
| OT in [24] | $O(\log n)$ | FullSim |
| ROT in [3] | 5 (including 2 with functionality) | UC-secure |
| Our ROT | 3 | UC-secure |

## References

1. Yao, A.C.: How to generate and exchange secrets. In: 27th Annual Symposium on Foundations of Computer Science, pp. 162–167 (1986)
2. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: Proceedings of the nineteenth annual ACM symposium on Theory of computing, pp. 218–229 (1987)
3. Costa, B., Branco, P., Goulao, M., Lemus, M., Mateus, P.: Randomized oblivious transfer for secure multiparty computation. Entropy **23**, 1001 (2021)
4. Yang, W., Huang, L.S., Wang, Q.Y., Luo, Y.L.: Quantum bit commitment based on qubit oblivious transfer. Chin. J. Electron. **18**(3), 422–426 (2009)
5. Yang, L.: Bit commitment protocol based on random oblivious transfer via quantum channel. arXiv: 1306.5863 (2013)
6. Song, Y.Q., Yang, L.: Practical quantum bit commitment protocol based on quantum oblivious transfer. Appl. Sci. **8**, 1990 (2018)
7. Pinkas, B., Rosulek, M., Trieu, N., Yanai, A.: SpOT-light: lightweight private set intersection from sparse OT extension. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11694, pp. 401–431. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_13
8. Pinkas, B., Rosulek, M., Trieu, N., Yanai, A.: SpOT-Light: lightweight private set intersection from sparse OT extension. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11694, pp. 401–431. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_13
9. Rabin, M.O.: How to Exchange Secrets by Oblivious Transfer. Technical Memo TR-81 (1981)
10. Aiello, B., Ishai, Y., Reingold, O.: Priced oblivious transfer: how to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 119–135. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44987-6_8
11. Camenisch, J., Neven, G., Shelat, A.: Simulatable adaptive oblivious transfer. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 573–590. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72540-4_33

12. Green, M., Hohenberger, S.: Blind identity-based encryption and simulatable oblivious transfer. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 265–282. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76900-2_16
13. Jannati, H., Bahrak, B.: An oblivious transfer protocol based on elgamal encryption for preserving location privacy. Wireless Pers. Commun. **97**(2), 3113–3123 (2017). https://doi.org/10.1007/s11277-017-4664-7
14. Ishai, Y., Kilian, J., Nissim, K., Petrank, E.: Extending oblivious transfers efficiently. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 145–161. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_9
15. Mahadev, U.: Classical Homomorphic Encryption for Quantum Circuits. SIAM J. Comput. 189 (2020)
16. Mahadev, U.: Classical Verification of Quantum Computations. In: 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS), pp. 259–267 (2018)
17. Alexandru, C., Léo, C., Elham, K., Petros, W.: On the possibility of classical client blind quantum computing. Cryptography **5**(1), 3 (2021)
18. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: ACM Symposium on Theory of Computing, 84–93 (2005)
19. Agrawal, S., Freeman, D.M., Vaikuntanathan, V.: Functional encryption for inner product predicates from learning with errors. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 21–40. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_2
20. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41
21. Canetti, R.: Universally composable security: a new paradigm for cryptographic protocols. In: IEEE Symposium on Foundations of Computer Science, p. 136 (2001)
22. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_31
23. Wang, F.H., Hu, P.Y., Liu, Z.H.: Lattice-based oblivious transfer protocol. J. Commun. **32**(3), 125–130 (2011)
24. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Adaptive oblivious transfer with access control from lattice assumptions. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 533–563. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_19

# A Secure Aggregation Scheme for Model Update in Federated Learning

Baolin Wang[1], Chunqiang Hu[1,2(✉)], and Zewei Liu[1]

[1] School of Big Data & Software Engineering, Chongqing University, Chongqing 400044, China
blwang@cqu.edu.cn, chu@cqu.edu.cn, zwliu@cqu.edu.cn

[2] Joint Laboratory on Cyberspace Security, China Southern Power Grid, Guangzhou, China

**Abstract.** Federated learning is a novel machine learning framework that effectively satisfies the requirements of multiple organizations for data usage and model training while meeting privacy protection, data security, and government regulations. However, recent research has shown that attackers can infer users' private information from their shared model parameters. To address the issue, in this paper, we propose the smart contract assisted secure aggregation scheme (SCSA). Firstly, we present a triple layers architecture based on blockchain for secure aggregation, which can adapt to application scenarios where a large amount of devices are involved in model training. Then, with the help of smart contracts, our scheme can efficiently distribute security masks to users in a decentralized form to ensure the security of parameters, and combine with secret sharing to design a double fault tolerance mechanism to effectively improve the robustness of the system. Finally, the theoretical analysis and simulation experiments prove that our scheme has high security and robustness while maintaining efficiency.

**Keywords:** Federated Learning (FL) · Secure aggregation · Smart contract · Communication-efficient · Robust

## 1 Introduction

Federated learning (FL) [1,2] is an emerging deep learning framework that meets the needs of safely training deep learning models on sensitive data sets while ensuring the privacy of data owners. Unlike traditional centralized model training, in FL, the data owner can utilize private data stored on a local device to train a full or partial model downloaded from a central server, instead of sending private data to the central server for model training. In order to accelerate the convergence of the model, the central server aggregates the local parameters (i.e., gradients) shared by each data owner and updates the global model [3].

Supported by the National Natural Science Foundation of China under grant 62072065.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 500–512, 2022.
https://doi.org/10.1007/978-3-031-19208-1_41

Recent studies indicate that FL also faces the risk of privacy leakage and security attacks [4,5]. On the one hand, the attacker can intercept the model and infer the private information contained in the model by launching inference attacks [6,7] and inversion attacks [8]. On the other hand, the attacker can indirectly acquire sensitive message such as data labels, memberships and features based on the shared gradient [4]. To address such security and privacy issues, the existing works integrate novel privacy-preserving techniques and encryption algorithm such as secret sharing, secure multi-party computing (SMPC), zero-knowledge proof (ZKP) and homomorphic encryption (HE) into FL to realize secure aggregation of parameters during the model update [9].

Recently, the rapid rise of blockchain technology has attracted a lot of attention. Based on it, combining blockchain to solve problems in FL is a hot research topic, such as model poisoning [10], edge computing [11], node trust and incentive [12]. Nevertheless, few studies have focused on addressing security and privacy breaches by leveraging blockchain features in the aggregation process [13,14]. Therefore, in this paper, we propose a novel federated learning triple layers architecture based on blockchain and present a secure aggregation scheme executed in a decentralized manner with the assistance of smart contracts. The contributions are summarized as follows:

- The paper presents the construction of a *cloud-edge-terminal* architecture for secure aggregation based on blockchain, in which all intermediate results of the aggregation process are stored to ensure that the aggregation results in no way can be tampered with and have the ability to track down malicious entities.
- The proposed scheme provides a double fault tolerance mechanism based on secret sharing and smart contracts, which enhances the robustness of the scheme while reducing the number of communications.
- Comprehensive theoretical analysis is provided to demonstrate the effectiveness of the scheme to handle the proposed attack model. In addition, we comprehensively evaluate the performance, robustness, and practicality of the scheme by simulating realistic scenarios.

The remainder of this paper is organized as follows. Section 2 introduces related works. Section 3 presents the system model and design goals. Section 4 briefly describes related preliminaries contained in our scheme. Section 5 shows the overall detailed workflow of SCSA. Sections 6 and 7 present the security and performance analyses, respectively. Finally, conclusions are drawn in Sect. 8.

## 2 Related Works

In this section, we compare the existing work based on the features in our system. Bonavitz *et al.* [15] introduced secure aggregation for FL, and proposed the first secure aggregation protocol. Inspired by Diffie-Hellman key exchange, they designed an algorithm to generate random mask values to obscure the real

parameters through multiple rounds of communication between users, and combined shamir secret sharing in the scheme to solve the issue of client dropout. Beguier *et al.* [16] developed a secure aggregation protocol among multiple servers, which utilizes data compression techniques to compress model updates before aggregation, with minimal communication overhead and computational cost. N. Dowlin *et al.* [17] proposed a scheme called CryptoNets, which allows homomorphically encrypted user data feedforwarding an already trained neural network. S. Nath *et al.* [18] applied threshold homomorphic encryption to encrypt the local parameters of each client before upload. The cloud server aggregates the ciphertext and jointly decrypts the ciphertext under the premise of meeting the threshold.

To sum up, the existing scheme still have disadvantages such as insufficient availability low efficiency, weak robustness. In addition, the above mentioned works generally adopt the *cloud-terminal* architecture, which is not scalable with the explosive growth of mobile terminals and fails to cope with the demands of a massive amount of devices involved in training. Therefore, it is an inevitable trend to build a secure aggregation scheme for model update in FL.

## 3 Problem Formalization

### 3.1 System Model

The smart contract assisted secure aggregation scheme consists of a key generating center, a cloud server, some edge servers and some mobile client groups that contain a number of clients, as shown in Fig. 1.

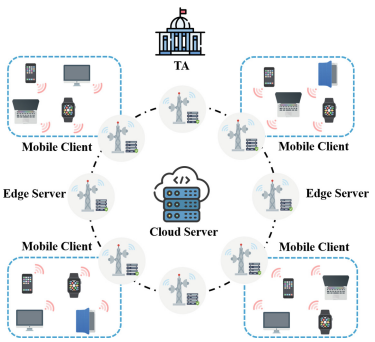



**Fig. 1.** System Model

- **Trusted Authority (TA):** The $TA$ is a fully trusted entity responsible for registering other entities. In addition, it generates the necessary public and private parameters, including the communication key for the registered entity.

- **Cloud Server (CS):** The $CS$ is responsible for sending the global model to the mobile client and updating the global model by aggregating the regional aggregation results of the edge server.
- **Edge Server (ES):** The $ES$ acts as consensus node to maintains the operation of the edge blockchain, and collects the updated parameters of all mobile clients in the specified region and conducting regional secure aggregation.
- **Mobile Client (MC):** The $MC$ can receive the global model from the cloud server, and use the locally stored data to train the model when the mobile client have sufficient power. After completing the local training process, MC sends the updated model parameters to the specified edge server.

### 3.2 Threat Model

Our scheme should be capable of meeting the following challenges

**Tampering Attacks:** During transmission, adversary can eavesdrop the communication channel between client and edge server and try to reveal the privacy information (i.e., data features) contained in the model parameters. Moreover, adversary is capable of launching active attacks such as false data injection.

**Collusion Attacks:** In each iteration of training, a malicious edge server $MS$ may collude with a subset of mobile client to invade the privacy of other clients. $MS$ can receive mask value from the clients with whom it colludes and uses these mask values to infer other clients' model parameters.

**Denial of Service (DoS) Attacks:** The external adversary consumes the resources of the edge server through sending frequent service requests to the edge server, causing the edge server to fail to provide services normally.

## 4 Preliminaries

### 4.1 Smart Contract and Blockchain

Smart contracts (SC) are programmable computer protocols that are disseminated, verified and executed in an informational way. It allows credible transactions without a third party and ensures that the transaction is traceable and irreversible. Blockchain (BC) is a novel application pattern of encryption algorithm, consensus mechanism, distributed data storage and other technologies. Its characteristics such as decentralization, openness, non-tampering, and traceability provide a credible execution environment for the implementation and application of smart contract [19].

### 4.2 Secret Sharing

In our scheme, we utilize the $t$-out-of-$N$ secret sharing protocol proposed by Shamir in [20]. Define a client set $C$, with $|C| = N$, a threshold value $t$ and a secret $s$. The Shamir's secret sharing protocol consists of a sharing algorithm $\mathbf{SS.share}(s, t, C) \rightarrow \{(i, s_i)\}_{i \in N}$, which can separates $s$ into $N$ pieces $s_1, s_2, ..., s_N$ and each secret piece $c_i$ is shared with the client in $C$. The secret sharing protocol ensures that any shared subset larger than $t$ can recover the secret $s$ by performing a reconstruction algorithm $\mathbf{SS.recon}(\{(i, s_i)\}_{i \in M}, t) \rightarrow s$ where $t \leqslant M \leqslant N$. If the number of secret pieces is less than $(t-1)$, however, the secret is not recovered in any case.

## 5 Our Schemes

In this section, we present SCSA for model update in FL. It consists of four phases: system initialization, scurity shield distribution, regional aggregation and global aggregation.

### 5.1 System Initialization

*TA* first generates public parameter and master secret key via $\mathbf{Setup}(1^\lambda) \rightarrow (PK,\ MSK)$, where $\lambda$ is a security parameter. The $CS$ provides unique identifier $ID_{CS}$ as the identity recognition to *TA* who will initiate the public/private key pair $(PK_{CS}, SK_{CS})$ of the $CS$ applying $\mathbf{KeyGen}(PK, MSK, ID_{CS}) \rightarrow (PK_{CS}, SK_{CS})$. $ES$ and $MC$ get key pair respectively via the same procedure.

### 5.2 Security Shield Distribution

In this phase, the $CS$ is responsible for generating global shield as well as local shield for each $ES$. The detailed process is described as follows.

***Broadcast***: Before security aggregation is required, the $CS$ broadcasts a secure aggregation request, which includes the basic information $\mathcal{M}$ of the model to be updated, some restrictions $\gamma$ that need to be satisfied, the hash value $H(\mathcal{M}, \gamma)$, and current timestamp $t$ to each $ES_i$, we denote it as $req = (\mathcal{M}, \gamma, H(\mathcal{M}, \gamma), t)$. Afterwards, the $CS$ encrypts the request message using the $PK_{ES_i}$ to get the ciphertext $REQ$ and sends it to the $ES_i$.

***Response***: After the $ES_i$ receives the broadcasted request, it decrypts the ciphertext $REQ$ using the $SK_{ES_i}$. If the decryption succeeds and the hash verification passes, the $ES_i$ accepts the request. Otherwise, the request is ignored. Then, the $ES_i$ confirms whether the limit condition $\gamma$ is satisfied. Finally, the $ES_i$ generates $res = (ID_i, addr_i, H(ID_i, addr_i), t')$ and sends $RES = \mathbf{Enc}(PK_{CS}, res)$ in response to the request to declare that it will participate in this aggregation.

***Verification***: For the received response messages, the $CS$ first confirms the validity of the reply by calculating $\|t - t'\| \leq \Delta t$. If the reply exceeds the

time limit, it will be ignored. Then $CS$ verifies the $ID_i$ to confirm whether the identity of the $ES_i$ is legal. If the verification is positive, $CS$ computes $h_i = H(ID_i||PK_{ES_i}||addr_i)$ and stores it in the candidate list $l = \{h_1, h_2, ..., h_n\}$.

***Distribution***: When $CS$ completes all request verifications, it selects a random values $\varepsilon$ as security parameter to generate the global shield $\boldsymbol{\xi}$ by calculating Eq. 1, then uses $\boldsymbol{\xi}$ and $l$ to invoke the security shield distribution protocol (SSDP) to assign a regional shield to each ES. The SSDP is shown in **Algorithm 1**.

$$\boldsymbol{\xi} = \mathbf{PRG}(\varepsilon) \tag{1}$$

where $\mathbf{PRG}(\varepsilon)$ is a pseudorandom generator with seed $\varepsilon$.

**Algorithm 1** Smart Contract of SSDP

**Input:** Candidate list $l = \{h_1, h_2, ..., h_n\}$, global shield $\boldsymbol{\xi}$.
**Output:** Regional shield list
1: **for** each $h_i$ in $l$ **do**
2: Get current system's timestamp $\phi$
3: Compute regional shield $\boldsymbol{shield_i} = \mathbf{PRG}(\phi)$
4: Update the global shield $\boldsymbol{\xi} - = \boldsymbol{shield_i}$
5: Store $\{h_i : \boldsymbol{shield_i}\}$ into BC. Here, $h_i$ is the index used to query the regional shield.
6: **end for**
7: **return** updated $\boldsymbol{\xi}'$ to CS

### 5.3 Regional Aggregation

In the regional aggregation, the edge server requires to communicate with all the mobile terminals in its jurisdiction. The process is similar to *Security Shield Distribution* phase, so only the differences are described here.

***Broadcast***: The $ES_i$ broadcasts request information $REQ = \mathbf{Enc}(PK_{MC_{i*}}, (\mathcal{M}, \gamma', H(\mathcal{M}, \gamma', t)))$ to all mobile clients in its jurisdiction.

***Response***: After the $REQ$ verification passes, if the $MC_{ij}$ decides to participate in this round of security aggregation then generates $res = (ID_{ij}, PK_{MC_{ij}}, addr_{ij}, H(ID_{ij}, addr_{ij}), t')$ and sends the message $RES = \mathbf{Enc}(PK_{ES_i}, res)$.

***Verification***: The $ES_i$ generates the list $pkl = \{PK_{MC_{i1}}, ..., PK_{MC_{in}}\}$ and the regional candidate list $l' = \{h'_1, h'_2, ..., h'_n\}$, with $h'_i = H(ID_{ij}||PK_{MC_{ij}}||addr_{ij})$.

***Masking generation***: After completing the above steps, $ES_i$ uses $h'_i$, $l$ and $h_i = H(ID_i||H(PK_{ES_i})||addr_i)$ to invoke the security mask distribution protocol (SMDP) to assign security masking to each $MC_{ij}$. The SMDP is shown in **Algorithm 2.**

The $MC_{ij}$ obtains encrypted security mask $\varsigma$ through invoking the smart contract $Query(h_i)$, and then utilizes the private key to decrypt the $\varsigma$ to obtain

**Algorithm 2** Smart Contract of SMDP

**Input:** Public key list $pkl = \{PK_{MC_{i1}}, PK_{MC_{i2}}, ..., PK_{MC_{in}}\}$, regional candidate list $l' = \{h'_1, h'_2, ..., h'_n\}$, hash value $h_i$ of $ES_i$.
**Output:** Security masking list

Invoke $Query(h_i)$, which is a smart contract with query function.
**if** query result is exist **then**
  get the regional shield $\boldsymbol{shield_i} = Query(h_i)$
  **for** $i = 0$ to $n$ **do**
    Get current system's timestamp $\phi$
    Compute security masking $\boldsymbol{mask_{ij}} = \mathbf{PRG}(\phi)$ for $MC_{ij}$
    Update regional shield $\boldsymbol{shield_i} -= \boldsymbol{mask_{ij}}$
    Encrypted security masking with $MC_{ij}$'s public key to get ciphertext $\varsigma = Enc(PK_{MC_{ij}}, mask_{ij})$
    Store $\{h'_i : \varsigma\}$ into BC. Here, $h'_i$ is the index used to query the ciphertext of security masking.
  **end for**
  Return updated $\boldsymbol{shield_i}'$ to $ES_i$
**else**
  Abort algorithm
**end if**

security mask $\boldsymbol{mask_{ij}} = \mathbf{Dec}(\varsigma, PK_{MC_{ij}})$. Next, The $MC_{ij}$ randomly selects a value $\varepsilon$ to generate the double masking $\boldsymbol{\epsilon_{ij}}$ through Eq. 1.

***Secret sharing:*** The $MC_{ij}$ sets a threshold $g$, selects a subset of mobile client $C = \{MC_{i1}, MC_{i2}, ..., MC_{ik}\}$, with $|C| = k \geqslant g$, in the region where it belongs. Then, according to the secret sharing protocol, $MC_{ij}$ executes $\mathbf{SS.share}(\boldsymbol{mask_{ij}}) \rightarrow \{(i, m_i)\}_{i \in k}$ and $\mathbf{SS.share}(\boldsymbol{\epsilon_{ij}}) \rightarrow \{(i, e_i)\}_{i \in k}$ separately to split $\boldsymbol{mask_{ij}}$ and $\boldsymbol{\epsilon_{ij}}$ into $k$ segments, and send $(m_i, e_i)$ to each $MC \in C$.

Ultimately, the $MC_{ij}$ masks the real parameters by calculating the Eq. 2 and sends $(H(ID_i||H(PK_{ES_i})||addr_i), \boldsymbol{y_{ij}}, H(\boldsymbol{y_{ij}}), t)$ to the $ES_i$.

$$\boldsymbol{y_{ij}} = \boldsymbol{x_{ij}} + \boldsymbol{mask_{ij}} + \boldsymbol{\epsilon_{ij}} \tag{2}$$

***Reconstruction and aggregation:*** The $ES_i$ receives the masked parameters from $MC_{ij}$ that is belong to $ES_i$'s administration region according to the candidate list. Then, $ES_i$ requests the secret fragment from the $MC \in C$ according to the parameter acceptance status. It requests the $m_i$ of $MC$ that did not send parameters due to dropout, and the $e_i$ of $MC$ that successfully sent parameters. We assume that the honest $MC$ only send one of $m_i$ and $e_i$ to $ES$. Next, $ES_i$ reconstructs $\boldsymbol{mask_{ij}}'$ and $\boldsymbol{\epsilon_{ij}}'$ respectively by executing $\mathbf{SS.recon}(\cdot)$. Finally, $ES_i$ completes regional aggregation by calculating the Eq. 3.

$$\begin{aligned} \boldsymbol{z_i} &= \sum_{j=0}^{n} \boldsymbol{y_{ij}} - \sum_{j=0}^{n} \boldsymbol{\epsilon_{ij}}' + \boldsymbol{shield_i}' \\ &= \sum_{j=0}^{n} (\boldsymbol{x_{ij}} + \boldsymbol{mask_{ij}} + \boldsymbol{\epsilon_{ij}}) - \sum_{j=0}^{n} \boldsymbol{\epsilon_{ij}}' + \boldsymbol{shield_i}' \\ &= \sum_{j=0}^{n} \boldsymbol{x_{ij}} + \sum_{j=0}^{n} \boldsymbol{mask_{ij}} + \boldsymbol{shield_i}' \\ &= \sum_{j=0}^{n} \boldsymbol{x_{ij}} + \boldsymbol{shield_i} \end{aligned} \tag{3}$$

### 5.4 Global Aggregation

After completing the regional aggregation, The $ES_i$ in each region sends the global aggregation message $M = (ID_i, \boldsymbol{z_i}, H(PK_{ES_i} || \boldsymbol{z_i}), addr_i, t)$, which includes the $ES$'s unique identifier, regional aggregation outcome, account address, hash value and current system's timestamp.

The $CS$ receives the regional aggregation results from the $ES$ according to the candidate list $l$, and then performs global aggregation according to Eq. 4.

$$\begin{aligned} \boldsymbol{p} &= \sum_{i=0}^{n} \boldsymbol{z_i} + \boldsymbol{\xi}' - \boldsymbol{\xi} \\ &= \sum_{i=0}^{n} (\sum_{j=0}^{n} \boldsymbol{x_{ij}} + \boldsymbol{shield_i}) + \boldsymbol{\xi}' - \boldsymbol{\xi} \\ &= \sum_{i=0}^{n} \sum_{j=0}^{n} \boldsymbol{x_{ij}} + (\sum_{i=0}^{n} \boldsymbol{shield_i} + \boldsymbol{\xi}') - \boldsymbol{\xi} \\ &= \sum_{i=0}^{n} \sum_{j=0}^{n} \boldsymbol{x_{ij}} \end{aligned} \tag{4}$$

### 5.5 Double Fault Tolerance Mechanism

The following two special cases that arise during the security aggregation process are considered: 1) The masking value ($\boldsymbol{mask_{ij}}$, $\boldsymbol{\epsilon_{ij}}$) of dropped MC during the regional aggregation phase cannot be reconstructed through secret sharing. 2) The ES is offline during the global aggregation phase and fails to send the regional aggregation results to the CS. Both cases cause unavailability of the parameter aggregation results, undermining the robustness of the system and wasting computational resources.

Therefore, we propose a double fault tolerance mechanism based on secret sharing and blockchain. When case 1) occurs, *the first fault tolerance mechanism* allows $ES_i$ to reconstruct the mask value of the offline $MC$ through the secret sharing algorithm **SS.recon(·)**, so as to eliminate its impact on the aggregation result. If the reconstruction of the mask value of the offline $MC$ fails or if case 2) occurs, *the second double fault tolerance mechanism* allows $CS$ to obtain the shield value $shield_i$ of the whole region by querying the smart contract $Query(\cdot)$ and removing the aggregation result $\boldsymbol{z_i}$ of the whole region, thus ensuring that the aggregation results of the normal region can be applied to the model update.

# 6 Theoretical Analysis

## 6.1 Avoid Tampering Attacks

A third party without a private key cannot obtain the content of the message. The communication information between entities will be verified by the hash value. Since the key participates in the hash operation, the adversary cannot initiate a tampering attack by forging the hash value of the message.

## 6.2 Avoid Collusion Attacks

Firstly, the masking value of the mobile client is encrypted and stored in the blockchain. Only the mobile client with the corresponding private key can decrypt the ciphertext to obtain the masking value, so the adversary cannot directly obtain the masking value of the mobile client. Secondly, the user will generate a private double mask. During the secret recovery phase, the honest user will only send one of $\boldsymbol{mask}_{ij}$ and $\boldsymbol{\epsilon}_{ij}$. Therefore, the edge server cannot obtain the updated parameters of the mobile client by launching a collusion attack.

## 6.3 Avoid Denial of Service Attacks

The request information contains the hash value of the session key. After receiving the request, the hash value of the session key will be verified. If the verification fails, the request will be ignored without waiting or responding. Hence, the adversary cannot successfully implement a denial of service attack.

# 7 Performance Evaluation

In this section, we will evaluate the computation overhead of CASA and prove the robustness by simulating different mobile clients dropout scenarios.

## 7.1 Experiment Setup

The experiments are conducted on servers equipped with Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz(16 CPUs), 2.9 GHz and 8 GB of RAM. The operating system is Ubuntu 20.04 LTS, which runs in the virtual machine software (VMware Workstation Pro) of Windows platform.

The federated learning process is simulated by *Pytorch* and the main model is a three-layer convolutional neural network, which is trained and tested with MNIST dataset. We set the threshold of secret sharing $t$ not less than 51% of the total number of shares and set the key length of symmetric encryption to 128bit. The above contents are implemented in *Python*. The *Ganache* is used to build an *Ethereum* test private chain environment locally, smart contracts are written using *Solidity* to write smart contracts, and a system prototype is built using the *Truffle* framework to verify the effectiveness of the scheme.

### 7.2 Performance Evaluation

The performance of four components in the scheme: initialization, shield and mask generation, secret sharing and secret reconstruction, is evaluated by using time consumption as performance evaluation metrics.

To begin with, we simulate the scenario of 5000 terminals participating in federated learning, and design three partition schemes: a) 10 regions with 500 mobile terminals in each region, b) 50 regions with 100 mobile terminals in each region, c) 100 regions with 50 mobile terminals in each region. d) A traditional cloud-client architecture is simulated using 5000 mobile clients located in the same region. Specifically, all terminals in each region are served by one edge server, so the number of edge servers is equal to the number of partitions. Considering that the time consumption of other parts of the schemes is negligible compared with the time consumption of assigning masks and mask values, only the time consumption of masks and mask generation is compared.

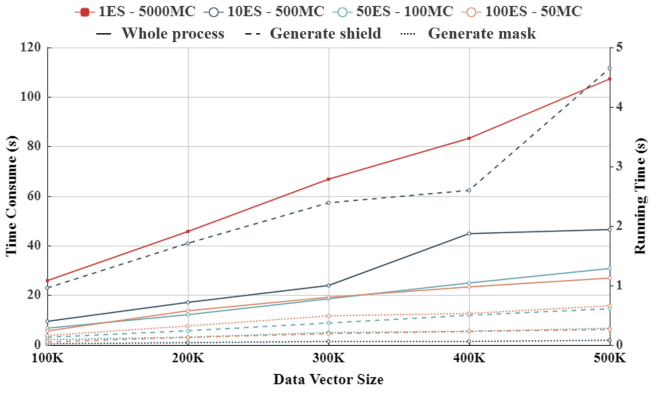



**Fig. 2.** Comparison of different partitioning schemes at different data vector sizes

Figure 2 indicates that all three partition schemes significantly outperform the traditional scheme for the same data vector size, and the whole process time consumes decreases as the number of partitions increases. This demonstrates that our proposed cloud-edge-client architecture can effectively improve the efficiency of mask value generation and distribution in the federated learning security aggregation process compared to the cloud-client architecture. The growth rate of the partitioning scheme is also significantly lower than that of the traditional scheme for different data vector size, so our scheme can accommodate the training of massive parameter models and is compatible with a large number of devices involved in federated learning. At the same time, the more regions are divided, the more time is consumed to generate shield, and the more terminals in the region, the more time is consumed to generate mask, so the reasonable division of the number of regions and devices in the region is essential.

### 7.3 Robustness Verification

Secret sharing and smart contracts are implemented in the proposed scheme to cope with the situation where the aggregation results are not available due to abnormal exit of the client during secure aggregation.

We set up experimental groups with mobile client numbers of 100, 200, 300, 400, and 500, and examined the efficiency of the secret sharing phase under the evaluation of data vector sizes of 100K, 200K, 300K, 400K, 500K. After that, the data vector size is fixed at 500K and the efficiency and communication overhead of secret reconstruction is measured for each experimental group at 0%, 10%, 20%, 30%, and 40% dropout rate.

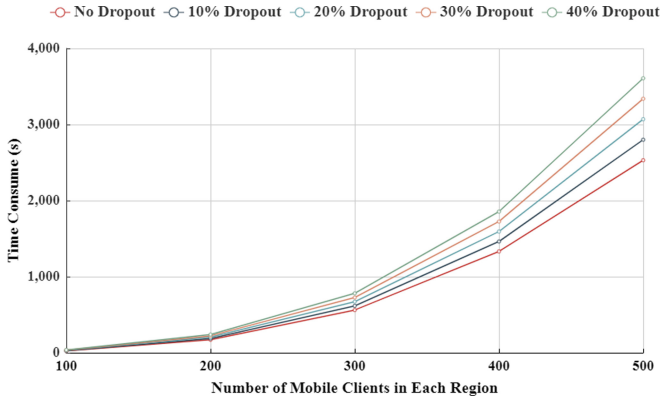



**Fig. 3.** The average running time of ES performing secret reconstruction with different MC dropout rates and different number of clients in a single region, the data vector size is fixed at 500K.

We measure the efficiency in coping with various dropout situations under different number of terminals. The experimental outcomes reveal that the time consumed for secret reconfiguration is proportional to the number of terminals with the same dropout rate. With the same number of clients, the higher the drop rate the longer it takes, but the increase is smaller, as shown in Fig. 3. This demonstrates that our scheme can efficiently reconstruct the lost information in response to massive user dropouts, which is sufficient to ensure the normal operation of secure aggregation.

## 8 Conclusion

In this paper, smart contract assisted security aggregation scheme SCSA is proposed, which consists of three layers: cloud service, edge layer and client layer. A blockchain deployed at the edge layer for recording intermediate data of the

security aggregation process. To improve the robustness of the system, a double fault tolerance mechanism based on secret sharing and smart contracts is designed to ensure that the secure aggregation process is minimally disturbed by dropout clients. Finally, the experimental results indicate that our scheme is highly feasible and robust.

## References


1. Mcmahan, H.B., Moore, E., Ramage, D., Arcas, B.A.Y.: Communication-efficient learning of deep networks from decentralized data. CoRR, vol. abs/1602.05629 (2016). [Online]. Available: http://arxiv.org/abs/1602.05629
2. Xiong, Z., Cai, Z., Takabi, D., Li, W.: Privacy threat and defense for federated learning with non-i.i.d. data in AIoT. IEEE Trans. Ind. Inform. **18**(2), 1310–1321 (2022)
3. Qin, Y., Kondo, M.: Mlmg: multi-local and multi-global model aggregation for federated learning. In: 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), pp. 565–571 (2021)
4. Peter Kairouz, H., McMahan, B., Avent, B., Bellet, A.: Advances and open problems in federated learning (2021)
5. Li, J., Cheng, S., Li, Y., Cai, Z.: Approximate holistic aggregation in wireless sensor networks. In: 2015 IEEE 35th International Conference on Distributed Computing Systems, pp. 740–741 (2015)
6. Nasr, M., Shokri, R., Houmansadr, A.: Comprehensive privacy analysis of deep learning: passive and active white-box inference attacks against centralized and federated learning. In: IEEE Symposium on Security and Privacy (SP) (2019)
7. Pang, J., Huang, Y., Xie, Z., Han, Q., Cai, Z.: Realizing the heterogeneity: a self-organized federated learning framework for IoT. IEEE Internet Things J. **8**(5), 3088–3098 (2021)
8. Ganju, K., Wang, Q., Yang, w., Gunter, C.A., Borisov, N.: Property inference attacks on fully connected neural networks using permutation invariant representations. In: ACM SIGSAC Conference, pp. 619–633 (2018)
9. Cai, Z., Xiong, Z., Xu, H., Wang, P., Li, W., Pan, Y.: Generative adversarial networks: a survey toward private and secure applications. ACM Comput. Surv. **54**(6), 1–38 (2021)
10. Short, A.R., Leligou, H.C., Papoutsidakis, M., Theocharis, E.: Using blockchain technologies to improve security in federated learning systems. In: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 1183–1188 (2020)
11. Deng, Y., Han, T., Zhang, N.: Flex: trading edge computing resources for federated learning via blockchain. In: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–2 (2021)
12. Xu, Y., et al.: BESIFL: blockchain empowered secure and incentive federated learning paradigm in IoT. IEEE Internet Things J. (2021)
13. Feng, L., Yang, Z., Guo, S., Qiu, X., Li, W., Yu, P.: Two-layered blockchain architecture for federated learning over the mobile edge network. IEEE Network **36**(1), 45–51 (2022)
14. Yuwen, P., Chunqiang, H., Deng, S., Alrawais, A.: $R^2$peds: a recoverable and revocable privacy-preserving edge data sharing scheme. IEEE Internet Things J. **7**(9), 8077–8089 (2020)

15. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., Seth, K.: Practical secure aggregation for federated learning on user-held data (2016)
16. Beguier C., Tramel, E.W.: SAFER: sparse secure aggregation for federated learning. CoRR abs/2007.14861 (2020)
17. Dowlin, N., Gilad-bachrach, R., Laine, K., Lauter, K., Wernsing, J.: CryptoNets: applying neural networks to encrypted data with high throughput and accuracy. IEEE (2016)
18. Nath S., Rastogi, V.: Private aggregation of distributed time-series data. US (2012)
19. Meng, T., Zhao, Y., Wolter, K., Cheng-Zhong, X.: On consortium blockchain consistency: a queueing network model approach. IEEE Trans. Parallel Dis. Syst. **32**(6), 1369–1382 (2021)
20. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)

# A Novel Self-supervised Few-shot Network Intrusion Detection Method

Jing Zhang[1,2(✉)], Zhixin Shi[1], Hao Wu[1,2], and Mengyan Xing[1,2]

[1] Institute of Information Engineering, Chinese Academy of Science, Beijing, China
{zhangjing2812,shizhixin,wuhao0010,xingmengyan}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Science, Beijing, China

**Abstract.** Supervised models for network intrusion detection usually rely on many training samples, but the annotation costs are very high. Unlabeled network traffic data is relatively easy to obtain. However, there are only a few methods to utilize these unlabeled data adequately. We propose a novel self-supervised few-shot network intrusion detection method to address the above problems. The method consists of two models: a) network traffic representation model and b) network intrusion detection model. First, the network traffic representation model uses unlabeled network traffic data through self-supervised learning to obtain network traffic representations, which will benefit the training of network intrusion detection model. Then, the shared layers of the network traffic representation model are transferred to the network intrusion detection model and frozen. Finally, a few training samples are used to fine-tune the network intrusion detection model, and we can obtain a model with good generalization. However, self-supervised learning of the network traffic representation model requires a method for generating labels from network traffic. Therefore, we propose a novel method to generate labels based on discrete features of network traffic. Experiments show that our proposed method has better performance than other network intrusion detection models with few-shot. On NSL-KDD, only 200 labeled samples are needed to achieve 95.2% accuracy.

**Keywords:** Self-supervised learning · Network intrusion detection · Generating labels · Few-shot learning

## 1 Introduction

The rapid development of computer networks and the internet of things has brought convenience to our lives. Still, at the same time, various forms of network attacks appeared [1]. How to identify and respond to these network attacks has become a current research hotspot. Network intrusion detection technology is one of the critical technologies for dealing with network attacks. It guarantees network security by detecting abnormal traffic [2]. Some network intrusion detection models based on traditional methods [3–5] and deep learning(DL) [6–11]

Supported by organization Zhixin Shi Youth Promotion Association.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 513–525, 2022.
https://doi.org/10.1007/978-3-031-19208-1_42

have been proposed in recent years. With a sufficient number of training samples, these models can achieve satisfactory results in network intrusion detection. A large amount of data annotation requires a lot of costs. And in some real-world scenarios, it is difficult for us to obtain enough training samples. For example, when a new vulnerability is discovered, the latest vulnerability is also known as a zero-day vulnerability. And there is no official patch for the vulnerability, there will be a large number of attacks that exploit this vulnerability, that is, zero-day attacks. In this scenario, the time to construct the dataset is minimal, and we cannot obtain a sufficient number of attack samples in a short period. The traditional network intrusion detection model that requires many training samples will become inappropriate. Therefore, we need a few-shot network intrusion detection model [12].

In general, DL-based models usually contain many trainable parameters. They require many labeled samples to achieve optimal performance [13], and a small number of training samples will lead to poor generalization ability. Unlabeled network traffic is usually readily available, but few network intrusion detection models currently exploit such unlabeled data. Therefore, we propose a few-shot learning method based on self-supervised learning, as shown in Fig. 1, consisting of the network traffic representation model and the network intrusion detection model. First, the network traffic representation model uses unlabeled traffic data and our proposed method to generate labels to complete self-supervised training, in which data representation can be learned [14]. Then, as shown in Table 1, we transfer the shared layers of the trained network traffic representation model to the network intrusion detection model. Finally, we use a small amount of manually annotated network traffic to fine-tune the task-specific layers of the network intrusion detection model to obtain a model with good generalization performance. The labels generation method we proposed uses a discrete feature as labels for network traffic to complete self-supervised learning. Since network traffic contains multiple discrete features, it is necessary to select a discrete feature that can enable the network traffic representation model to learn better network traffic representations. Because the network traffic representation learned by the network traffic representation model is to improve the performance of the network intrusion detection model, we directly select a discrete feature as labels by observing the performance of the network intrusion detection model.

We propose a few-shot network intrusion detection method based on self-supervised learning to solve above problems. The advantages of our method are a) The network traffic representation model utilizes unlabeled network traffic data, and the network traffic representation obtained through self-supervised learning can improve the performance of network intrusion detection; b) Transfer shared layers of the network traffic representation model to the network intrusion detection model, and then only a small number of manually labeled samples are needed for fine-tuning. The contributions of this paper can be summarized as follows:

- We propose a novel self-supervised few-shot network intrusion detection method. As far as we know, we are the first to use free semantic label-based methods [14] to complete self-supervised learning of network traffic. Our model and the existing three shallow learning algorithms, XGBoost-DNN [6] all use a small number of samples from the NSL-KDD dataset as the training set, and our model has better performance on the same test set.
- We propose a new labels generation method to accomplish better self-supervised learning, which selects a discrete network traffic feature to generate labels. And we also demonstrate through experiments that the network traffic representation obtained through self-supervised learning can improve the performance of our network intrusion detection model on few-shot.
- To provide a reproducible model, we conduct a detailed analysis of the relevant parameters of the network intrusion detection method.

## 2 Related Works

As an essential part of network security, network intrusion detection has been paid more and more attention by researchers. Khan [15] proposed a network intrusion detection model based on a Convolutional Neural Network (CNN) algorithm (2019). Zhang [16] proposed a model combining Multiscale CNN with Long Short-Term Memory (2020). Yang [3] proposed a machine learning framework based on XGBoost and deep neural networks (2021). The above models have achieved satisfying results on large labeled datasets but have not effectively solved the few-shot problem.

In recent years, researchers proposed some few-shot intrusion detection models. Xu [2] proposed a few-shot network intrusion detection method based on a meta-learning framework that can learn prior knowledge for network traffic classification directly from original traffic (2020). The first step of the meta-learning framework still requires many labeled historical data. The second step is to use the newly collected data for fine-tuning. Yu [17] proposed a balanced resampling method and DL-based feature extraction, using similarity measure for intrusion detection, 1% of NSL-KDD KDDTrain+dataset for training, accuracy can reach 92.34% (2020). Although this method can reduce the number of training samples, 1% of KDDTrain+ still has 1259 samples, the cost of sample labeling is still high.

To avoid the labeling cost of large-scale datasets and make full use of unlabeled data, a concept of self-supervised learning has been proposed in computer vision [14]. Self-supervised learning uses unlabeled data to learn data representation and fine-tune it with a small amount of labeled data. It has achieved good results in image recognition with few samples. However, for self-supervised learning, how to generate labels from unlabeled data becomes a difficulty of this method. For sequential data, Sarkar [18] used the spatiotemporal transformation of ECG data to create labels to complete self-supervised training (2020). The network traffic data contains discrete features, and the method of spatiotemporal transformation is not suitable. Therefore, our first problem is to propose a method of generating labels for unlabeled network traffic.

## 3 Background

Self-supervised learning refers to learning methods in which ConvNets are explicitly trained with automatically generated labels [14]. The training data for supervised learning usually consists of data pairs $(x_i, y_i)$, where $i \in [1, N]$ and $N$ represent the dataset size. In general, $x_i$ represents the data features, and $y_i$ is the manual annotation. Self-supervised learning also requires a set of data pairs $(x_i, p_i)$. Unlike $y_i$, $p_i$ can be generated by semi-automatic data processing, or it can be part of the data itself. In the field of self-supervised learning, $p_i$ is often called pseudo label.

Self-supervised learning generally consists of pretext task $T_p$ and downstream task $T_d$. $T_p$ is a pre-defined network that uses $(x_i, p_i)$ as training data to learn data representations. The goal of $T_d$ is to predict $y_i$ through $x_i$. When the training data is insufficient, the network parameters of $T_d$ can be initialized from the network parameters learned from $T_p$. When $T_p$ learns the data representation, only a tiny amount of $(x_i, y_i)$ is required for fine-tuning in $T_d$.

From the above introduction, $T_p$ needs to learn a deep supervised model $M_{\theta_p}(x_i)$ to predict $p_i$, where $\theta_p$ is the set of trainable parameters. $T_d$ designs a new deep supervised model $M_{\theta_d}(x_i)$ by intercepting a part of $M_{\theta_p}(x_i)$ and adding new layers to predict $y_i$, in which the parameter $\theta_{p'}$ of the intercepted part of $M_{\theta_p}(x_i)$ will be directly transferred to $M_{\theta_d}(x_i)$. In the next training process $\theta_{p'}$ will not be changed.

In this paper, pseudo labels are constructed using unlabeled network traffic data. $T_p$'s model is the network traffic representation model, and the model corresponding to $T_d$ is the network intrusion detection model.

## 4 Proposed Method

This paper aims to use few-shot to classify normal traffic and abnormal traffic. Our method is divided into the following three steps to accomplish this goal.

### 4.1 Network Traffic Generation Pseudo Labels

For self-supervised learning, a critical problem is generating pseudo labels. Inspired by pseudo labels that can be obtained from the data itself by using a “semi-automatic" process [19], we can generate pseudo labels using some discrete features of the network traffic data itself. For example, network traffic data contains protocol feature. Suppose the protocol feature has $n$ kinds of protocols, and we will select these $n$ kinds of protocols as labels for network traffic. Network traffic contains multiple discrete features, so we need to select a discrete feature as labels that can make $T_p$ better learn network traffic representation.

Because $T_p$ learns a better network traffic representation will be more beneficial to the training of $T_d$, we choose a discrete feature to generate pseudo labels by observing the performance of $T_d$. Our proposed metric of performance

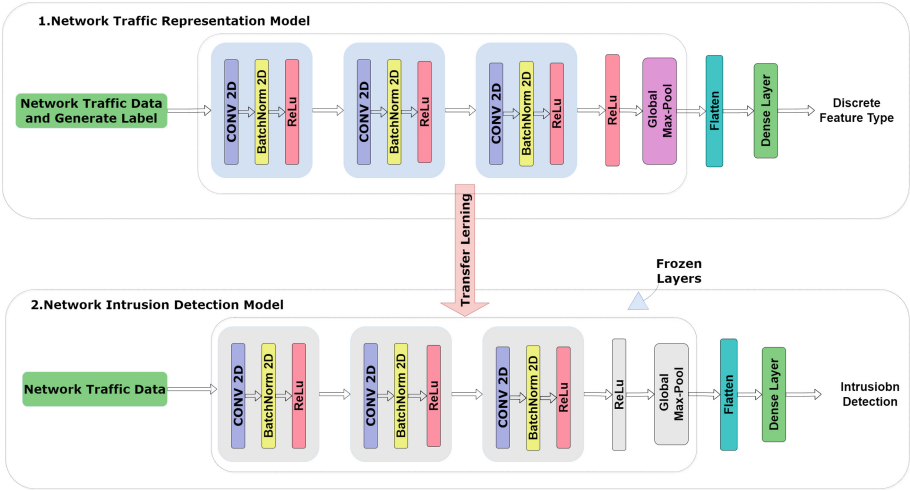



**Fig. 1.** This picture is the architecture of the few-shot learning method based on self-supervised learning proposed in this paper.

is shown in Eq. 1, where $Score_j$ corresponds to the performance of the $j$th discrete feature on $T_d$. First, we use different discrete network traffic features to generate pseudo labels to train a network traffic representation model. Then, we migrate the shared layers of different network traffic representation models to network intrusion detection models and use the same training set to fine-tune the network intrusion detection models, respectively. Finally, we observe the $Score_j$ of different network intrusion detection models on the same test set to generate pseudo labels. All in all, unlabeled network traffic will select the feature with the largest $Score_j$ to generate pseudo labels

$$Score_j = \frac{Accuracy_j + Precision_j + Recall_j + F1_j}{4} \tag{1}$$

### 4.2 Network Traffic Representation Model

The model used by $T_p$ is the network traffic representation model. The training data of the model is $(x_i, p_i)$. As shown in Table 1, the network traffic representation model consists of input, shared, task-specific, and output layers. Usually, we think of operations such as convolutional layers, pooling layers, and activation function layers as mapping the original data to the hidden layer feature space, that is, to obtain a better data representation. The connection layer usually plays the role of classification in CNN. To transfer the shared layers of the network traffic representation model to the network intrusion detection model, the latter model will perform better. We combine convolutional layers, pooling layers, and activation functions into shared layers, and task-specific layers consist of fully connected layers.

**Table 1.** The parameter settings of the network traffic representation model and the network intrusion detection model.

| Module | Layer Details | Feature Shape |
|---|---|---|
| Input | - | $7 \times 7$ |
| Shared Layers | [Conv2d, 6, 3, 1, 1] [BatchNorm2d,6] [ReLU] | $7 \times 7 \times 6$ |
| | [Conv2d, 16, 3, 1, 1] [BatchNorm2d,16] [ReLU] | $7 \times 7 \times 16$ |
| | [Cov2d, 32, 3, 1, 0] [BatchNorm2d,32] [ReLU] | $5 \times 5 \times 32$ |
| | [MaxPool2d, 2 , 2] | $2 \times 2 \times 32$ |
| Task-Specific Layers | [linear, 128, 512] [ReLU] [Dropout, 0.4] | 512 |
| Output | [linear, 512, $n$] | $n$ |

The shared layer has three convolutional blocks, each consisting of Cov2d, BatchNorm2d, and ReLu. In Cov2d layers, we gradually increase the number of filters from 6 to 16 and 32. The kernel size is $3 \times 3$ in all Cov2d layers. At the end of the shared layers, we used MaxPool2d, the kernel size is $2 \times 2$, and the stride is 2. The subsequent task-specific layers are composed of two fully connected layers, where the number of hidden layer nodes is 512. The number of output layer nodes is set according to the type of pseudo labels, and the learning rate and training batch size are 0.01 and 128, respectively. Both the network traffic representation model and the network intrusion detection model are used to complete the classification task, and the use of the cross-entropy loss function will facilitate the training of the classification model. Therefore, both models use the following cross-entropy loss function.

$$L\left(P_i, t_i\right) = t_i \log\left(P_i\right) + \left(1 - t_i\right) \log\left(1 - P_i\right) \tag{2}$$

The objective function of the network traffic representation model is shown in Function 3, where $\lambda = 0.001$, $\theta = \theta_p$, $t_i = p_i$. Each epoch inputs a batch of training data, calculates the objective function corresponding to the training data, and then updates the parameters through backpropagation technology. This is similar to traditional DL training, where we utilize Adam optimization method based on Stochastic Gradient Descent (SGD).

### 4.3 Network Intrusion Detection Model

The model used by $T_d$ is the network intrusion detection model, whose training data is $(x_i, y_i)$. Usually, the shared layers of CNN can learn data representation, and the task-specific layers are to complete specific classification tasks. We transfer the shared layers of the network traffic representation model learned above to the network intrusion detection model, and add the same task-specific layers as the network traffic representation model. It should be noted here that when training the network intrusion detection model, we did not change the parameters of the shared layers. We only trained its task-specific layers, thus greatly reducing the training cost of the network.

We intentionally keep the task-specific layers simple to evaluate whether $T_p$ learns network traffic representations better. Like the network traffic representation model, the training learning rate and batch size are 0.01 and 128, respectively. The objective function of the network intrusion detection model is shown in Function 3, where $\lambda = 0.001$, $\theta = \theta_d$, $t_i = y_i$. The Adam optimization method is also used here.

$$J(\lambda, \theta, t_i) = -\frac{1}{N}\sum_{i}^{N} L(P_i, t_i) + \frac{1}{2}\lambda \|\theta\|_2^2, P_i = M_\theta(x_i) \tag{3}$$

## 5 Experiments and Results

This section first introduces our chosen dataset, data preprocessing method, and experimental results. Then, we design three experiments to verify the effectiveness of our method. The first experiment uses our proposed pseudo labels generation method, which selects the discrete feature that can better learn network traffic representation to generate pseudo labels. The second experiment demonstrates that self-supervised learning of network traffic can improve the performance of our model on few-shot. In the third experiment, we will compare with other intrusion detection models on the same test set using a small number of training samples. The details of the datasets used for $T_p$ and $T_d$ in the three experiments are shown in Table 2. The training set is sampled from the dataset without replacement. Our experiments are repeated 15 times, the train and test sets in $T_p$ and $T_d$ are resampled each time, and the average of the 15 results is taken as the final result. Our proposed model is implemented using Pytorch, and the model runs on a 64-bit Centos with 16 GB RAM and a GeForce RTX 2080Ti.

**Table 2.** In the three experiments, the name of the dataset subset used by $T_p$ and $T_d$ and the usage of test and training sets.

| | | Tp | | Td | |
|---|---|---|---|---|---|
| Dataset | Experiment | Subset | Train | Subset | Train/Test |
| NSL-KDD | 5.3 | KDDTest | 5% | KDDTrain | 200/5000 |
| | 5.4 | | 80% | | - |
| | 5.5 | | 80% | | 1000/500 |
| UNSW-NB15 | 5.3 | UNSW_NB15_ testing-set | 8% | UNSW_NB15 (1–4) | 3%/5% |
| | 5.4 | | 80% | | - |

### 5.1 Datasets Description

To verify the effectiveness of our proposed method, we selected the mainstream intrusion detection datasets NSL-KDD and UNSW-NB15 for verification. NSL-KDD is generated based on KDD-Cup'99 [20]. NSL-KDD is to solve the problem that KDD-Cup'99 contains many redundant and duplicate records, its data volume is less than KDD-Cup'99, and these data points are all unique [16]. UNSW-NB15 was published in 2015, including nine different modern attack types and a wide variety of actual normal activities, and 49 features inclusive of the class label [21].

### 5.2 Data Preprocessing and Evaluation Metrics

To complete the model's training, we need to perform the following preprocessing. First, we encode discrete features. The encoding method we choose is an ordered encoder. This encoding method is straightforward to understand. All the features of the same category are encoded into the same value. Specifically, it converts discrete data into numbers between 0 and $n-1$, where $n$ is the number of all different categories of a feature. Then, to make the features of different dimensions in the same numerical order, reduce the influence of the features with large variance, and speed up the convergence speed of the learning algorithm, we perform min-max normalization on the encoded data. Finally, fill the encoding and normalization into a $7 \times 7$ matrix, and fill the insufficient with 1. It should be emphasized here that for the $T_p$, the discrete features used to generate the pseudo labels will all be set to 1 before preprocessing.

**Table 3.** The experimental results of $T_d$ on the test set after pseudo labels are constructed with different discrete features on the NSL-KDD dataset.

| Feature | Accuracy | Precision | Recall | F1 | Score |
|---|---|---|---|---|---|
| flag | 0.928 | 0.942 | 0.937 | 0.936 | 0.936 |
| is_guest_login | 0.858 | 0.834 | 0.847 | 0.843 | 0.846 |
| is_host_login | 0.904 | 0.963 | 0.823 | 0.888 | 0.895 |
| land | 0.847 | 0.852 | 0.827 | 0.834 | 0.840 |
| logged_in | 0.866 | 0.904 | 0.952 | 0.912 | 0.909 |
| protocol_type | 0.936 | 0.928 | 0.966 | 0.931 | 0.940 |
| root_shell | 0.902 | 0.921 | 0.863 | 0.887 | 0.893 |
| su_attempted | 0.906 | 0.943 | 0.892 | 0.898 | 0.910 |
| service | **0.952** | **0.964** | **0.972** | **0.945** | **0.958** |

The $T_d$'s network intrusion detection model can be regarded as a binary classification task, which will be measured using accuracy, precision, recall, and F1.

### 5.3 Pseudo Labels Generation

Based on NSL-KDD and UNSW-NB15, we will use different discrete features of network traffic data to generate pseudo labels, and train the models of $T_p$ and $T_d$ with the settings of the datasets in Table 2 respectively. In the $T_d$ task, NSL-KDD uses 200 samples as the training set and 5000 samples as the test set. In the $T_d$ task, UNSW-NB15 uses 3% of UNSW_NB15 (1–4) samples as the training set and 5% as the test set. $T_p$ is trained for 100 epochs, while the $T_d$ is trained for 250 epochs. The score is calculated by $T_d$'s accuracy, precision, recall and F1 on the test set, and the discrete feature corresponding to the highest score is selected to generate labels.

**Table 4.** The experimental results of $T_d$ on the test set after pseudo labels are constructed with different discrete features on the UNSW-NB15 dataset.

| Feature | Accuracy | Precision | Recall | F1 | Score |
|---|---|---|---|---|---|
| is_ftp_login | 0.853 | 0.840 | 0.906 | 0.872 | 0.867 |
| is_sm_ips_ports | 0.859 | 0.846 | 0.910 | 0.876 | 0.873 |
| service | 0.873 | 0.877 | 0.896 | 0.896 | 0.886 |
| state | 0.876 | 0.865 | 0.892 | 0.891 | 0.881 |
| proto | **0.921** | **0.950** | **0.919** | **0.913** | **0.926** |

By observing Table 3 and Table 4, we can see that the discrete feature service in NSL-KDD and the discrete feature proto in UNSW-NB15 generate pseudo labels and train $T_p$, $T_d$ can obtain the highest score on the test set, and the accuracy, precision, recall, and F1 are also the highest. The network traffic representation learned by $T_p$ is to improve the performance of $T_d$, and the performance on $T_d$ can reflect the quality of the network traffic representation learned by $T_p$. Therefore, we can conclude that the labels generated by discrete feature service and proto in these two different datasets are the most beneficial for $T_p$ to learn network traffic representation.

### 5.4 Improved Model Performance on Few-shot

In this part, we will verify whether the network traffic representation learned by $T_p$ can improve the performance of our model on few-shot. Our model will select features with the highest score in Table 3 and Table 4 to generate pseudo labels. NSL-KDD and UNSW_NB15 select service and proto, respectively. In this part, our model will be compared with CNN and Fully Connected Neural Network (FCNN) by training $T_d$, CNN, and FCNN using different ratios of training and test sets. To verify that the network traffic representation learned by $T_p$ can improve our performance on small samples, we set the structure of CNN to be the same as that of the network intrusion detection model, and the structure of FCNN is the same as the task-specific layers in the Table 1. The

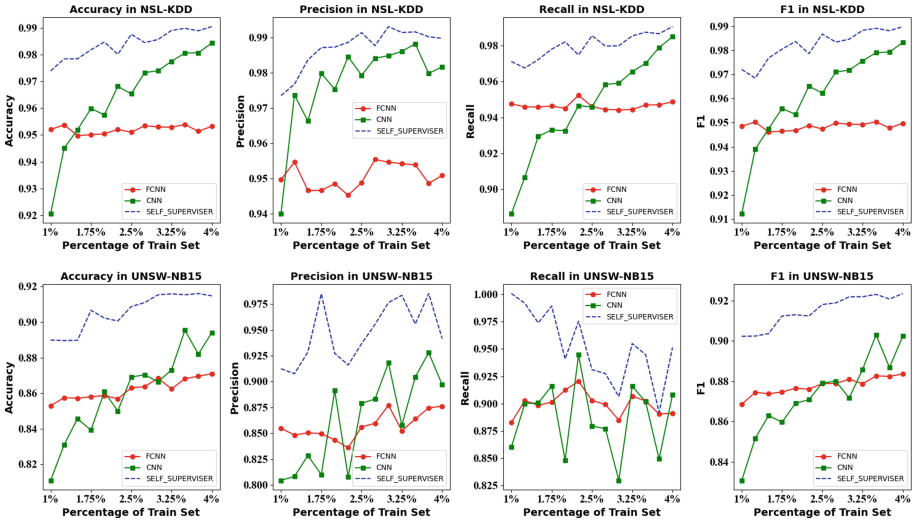



**Fig. 2.** The experimental results are obetained by using NSL-KDD and UNSW-NB15 respectively and changing the proportion of training data in $T_d$ under the same $T_p$.

training set used in $T_p$ is fixed, and the proportion of the training set of $T_d$ in the dataset is increased, the remaining data is used as the test set. The ratio is in the range of 1% to 4%, and each time it increases by 0.25%. $T_p$ is trained for 40 epochs in this experiment, while the $T_d$, CNN, and FCNN are trained for 60 epochs.

The experimental results are shown in Fig. 2. When using $1\% - 4\%$ of subsets of NSL-KDD and UNSW-NB15 as the training set, our proposed model outperforms CNN and FCNN on NSL-KDD and UNSW-NB15. CNN has the same structure as the network intrusion detection model, and the parameters of the three convolution blocks of CNN are updated during the training process. When training the network intrusion detection model, we only update the parameters of the task-specific layers. And the structure of FCNN is the same as the task-specific layers of the network intrusion detection model, which further shows that under the same network structure, the network traffic representation is learned through shared layers, which can improve the performance of the network intrusion detection model. Therefore, we can conclude that the network traffic representation obtained by $T_p$ through self-supervised learning can improve the learning performance of our model on few-shot.

**Table 5.** Comparison of our proposed model with other models

| Model | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| LR | 0.91 | 0.91 | 0.91 | 0.87 |
| NB | 0.85 | 0.85 | 0.85 | 0.85 |
| SVM | 0.95 | 0.81 | 0.70 | 0.66 |
| XGBoost-DNN | 0.97 | 0.94 | 0.94 | 0.97 |
| Our Model | **0.972** | **0.97** | **0.95** | **0.973** |

### 5.5 Network Intrusion Detection Model Comparison

In this part, our proposed method will be compared with other existing models on the NSL-KDD dataset. The comparison models we choose are Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM), and XGBoost-DNN [6]. Both $T_d$ and the above models use 1000 labeled samples as the training set and 5000 samples as the test set. Performance comparisons are made by observing accuracy, precision, recall, and F1. The $T_p$ is trained for 100 epochs, while the $T_d$ is trained for 250 epochs. As we can be seen from the Table 5, our model outperforms other network intrusion detection models, especially in precision and recall. The comparison between our proposed method and other models proves our method's effectiveness. It shows that the representation of network traffic obtained by self-supervised learning will benefit network intrusion detection on few-shot.

## 6 Conclusion

With respect to the high cost of a large amount of network traffic annotation, we propose a few-shot network intrusion detection method based on self-supervised learning. This method is divided into two stages. The first stage uses unlabeled network traffic data to obtain network traffic representation through self-supervised learning. The second stage uses the network traffic representation learned in the first stage and fine-tunes it with a small number of manually labeled samples to get the network intrusion detection model. Since self-supervised learning usually requires unlabeled samples to generate labels in a non-manual way, we also propose a method to generate labels for network traffic data. This method can generate the most beneficial labels for learning network traffic representations. Based on the mainstream datasets NSL-KDD and UNSW-NB15, three experiments are designed. The first experiment adopts the discrete feature which is best for learning traffic representations to generate labels. The second experiment demonstrates that our learned network traffic representation will be helpful for the training of network intrusion detection models. The third experiment demonstrates that our model outperforms other models. We can conclude that our proposed few-shot network intrusion detection method is novel and effective.

## References


1. Wang, L., Huang, W., Lv, Q., Wang, Y., Chen, H.Y.: AOPL: attention enhanced oversampling and parallel deep learning model for attack detection in imbalanced network traffic. In: Liu, Z., Wu, F., Das, S.K. (eds.) WASA 2021. LNCS, vol. 12938, pp. 84–95. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86130-8_7
2. Xu, C., Shen, J., Du, X.: A method of few-shot network intrusion detection based on meta-learning framework. Trans. Inf. Forensics Secur. **15**, 3540–3552 (2020)
3. Yang, T.-H., Lin, Y.-T., Wu, C.-L., Wang, C.-Y.: Voting-based ensemble model for network anomaly detection. In: ICASSP, pp. 8543–8547. IEEE (2021)
4. Xu, H., Przystupa, K., Fang, C., Marciniak, A., Kochan, O., Beshley, M.: A combination strategy of feature selection based on an integrated optimization algorithm and weighted k-nearest neighbor to improve the performance of network intrusion detection. Electronics **9**(8), 1206 (2020)
5. Gu, J., Lu, S.: An effective intrusion detection approach using SVM with naïve bayes feature embedding. Comput. Secur. **103**, 102158 (2021)
6. Devan, P., Khare, N.: An efficient XGBoost-DNN-based classification model for network intrusion detection system. Neural Comput. Appl. **32**(16), 12499–12514 (2020). https://doi.org/10.1007/s00521-020-04708-x
7. Zhang, H., Li, Y., Lv, Z., Sangaiah, A.K., Huang, T.: A real-time and ubiquitous network attack detection based on deep belief network and support vector machine. J. Autom. Sinica **7**(3), 790–799 (2020)
8. Sun, P., et al.: DL-IDS: extracting features using CNN-LSTM hybrid network for intrusion detection system. Secur. Commun. Netw. **2020**, 1–11 (2020)
9. Andresini, G., Appice, A., Malerba, D.: Autoencoder-based deep metric learning for network intrusion detection. Inf. Sci. **569**, 706–727 (2021)
10. Yang, Z., Leng, L., Zhang, B., Li, M., Chu, J.: Two novel style-transfer palmprint reconstruction attacks. Appl. Intell. 1–18 (2022)
11. Yang, Z., Xia, W., Lu, Z., Chen, Y., Li, X., Zhang, Y.: Hypernetwork-based personalized federated learning for multi-institutional CT imaging. arXiv preprint arXiv:2206.03709 (2022)
12. Tang, R., et al.: Zerowall: detecting zero-day web attacks through encoder-decoder recurrent neural networks. In: INFOCOM, pp. 2479–2488. IEEE (2020)
13. Jia, S., Jiang, S., Lin, Z., Li, N., Xu, M., Yu, S.: A survey: Deep learning for hyperspectral image classification with few labeled samples. Neurocomputing **448**, 179–204 (2021)
14. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: a survey. Trans. Pattern Anal. Mach. Intell. **43**(11), 4037–4058 (2020)
15. Khan, R.U., Zhang, X., Alazab, M., Kumar, R.: An improved convolutional neural network model for intrusion detection in networks. In: CCC, pp. 74–77. IEEE (2019)
16. Zhang, J., Ling, Y., Fu, X., Yang, X., Xiong, G., Zhang, R.: Model of the intrusion detection system based on the integration of spatial-temporal features. Comput. Secur. **89**, 101681 (2020)
17. Yu, Y., Bian, N.: An intrusion detection method using few-shot learning. IEEE Access **8**, 49730–49740 (2020)
18. Sarkar, P., Etemad, A.: Self-supervised ECG representation learning for emotion recognition. Trans. Affect. Comput. (2020)
19. Liu, X., et al.: Self-supervised learning: generative or contrastive. Trans. Knowl. Data Eng. (2021)

20. Wang, Z., Li, Z., Wang, J., Li, D.: Network intrusion detection model based on improved BYOL self-supervised learning. Secur. Commun. Netw. **2021**, 9486949 (2021)
21. Dwivedi, S., Vardhan, M., Tripathi, S.: Incorporating evolutionary computation for securing wireless network against cyberthreats. J. Supercomput. **76**(11), 8691–8728 (2020). https://doi.org/10.1007/s11227-020-03161-w

# A Trust Secure Data Aggregation Model with Multiple Attributes for WSNs

Zhaowei Li[1], Na Dang[1], Wenshuo Ma[2(✉)], and Xiaowu Liu[1]

[1] School of Computer Science, Qufu Normal University, Rizhao 276800, China
Lizw1996@foxmail.com
[2] School of Information Technology, Qingdao Vocational and Technical College of Hotel Management, Qingdao 266100, China
weimws@foxmail.com

**Abstract.** Wireless Sensor Networks (WSNs) are composed of many resource-limited nodes which may be laid in an unattended way. As a result, the sensing data in the transmission mechanism are sensitive to attacks launched by adversaries. In this paper, we propose a novel Trust Secure Data Aggregation Model (TSDAM) with multiple attributes for WSNs. Firstly, we calculate the direct trust based on the data accuracy, the energy consumption and the forwarding behavior of nodes. Secondly, the indirect trust is evaluated according to the communication behavior and the recommended credibility of neighbor nodes. Finally, the comprehensive trust is generated depending on various trusts, such as the direct and the indirect trust. Different from other mechanisms, TSDAM also selects the trust path according to the self-recommendation which is an attribute to indicate the willingness whether a node hope to participate in the communication process or not. The simulations show that TSDAM not only improves the reliability of the relay node, but also promotes the efficiency and accuracy of data aggregation.

**Keywords:** WSNs · Self-recommendation · Direct trust · Indirect trust

## 1 Introduction

Many trust models are introduced into WSNs in order to protect WSNs from being attacked by potential attacks and distinguish the credible nodes from the damaged ones [1–3]. Based on this idea, the trust-based Data Aggregation (DA) is proposed to enhance the security of WSNs. The prime consideration of designing related strategies is how to choose an optimal Aggregation Node (AN) of DA according to the trust value. In addition, some trust models are presented to improve the quantity of DA [4–8]. However, current trust models remain to be promoted and some challenging issues focus on the following aspects. i) Some models regard the factors including data, communication and energy as a reference for trust evaluation [9–14]. However, most of these models ignore the recommendation as a direct standard for trust evaluation, such as ADCT [7], LDTS [10] and TRPM [11]. ii) In general, the existing trust-based DA models rarely consider the self-recommendation of the node. iii) Some existing studies allow DA mechanism


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 526–534, 2022.
https://doi.org/10.1007/978-3-031-19208-1_43

to re-select a new AN when the current AN is attacked or even disabled by the enemy [13]. However, such DA models do not analyze the underlying causes of the current aggregating failure, which may limit the update speed of the trust system.

For the above problems in the DA model, this paper puts forward a novel Trust Secure Data Aggregation Model (TSDAM) based on multiple attributes, which has three contributions to improve the trust evaluation of WSNs. i) In TSDAM, the recommendations, derived from the neighboring nodes, are proposed in order to strengthen the robustness of trustworthiness evaluation. ii) We enrich TSDAM with self-recommendation which takes the willingness of node into consideration. iii) If the trust of current AN is below a predefined threshold in DA, TSDAM can adjust the current AN by means of the trust message received from intermediate nodes and BS.

The structure of this paper is arranged as follows. we discussed our model in Sect. 2. Section 3 presents the simulation experiments. Section 4 is mainly to summarize the paper and propose the further work.

## 2 Trust Secure Data Aggregation Model with Multiply Attributes

### 2.1 Network Model

In our model, the nodes are clustered at the initial stage of the network based on their one-hop distance and adjacent relationship. We assume that the network is secure during the initial deployment phase and each node has a unique identification (ID). The behaviors of Cluster Members (CMs) are monitored by their neighbor nodes.

### 2.2 Direct Trust(DT)

For easy representation, we use A and B to represent CH and CM, respectively.

**Data Trust.** Assumed that a set with $k$ elements, $S$, includes the sensing data perceive by $k$ neighbor nodes in a cluster. A element in set $S$, $s_i \in S$, is one of the sensing data and the average of all data is $\xi = 1/k \sum_{j=0}^{k} s_j$. Then, we can evaluate the deviation between the sensing data $s_i$ and the mean value $\xi$ and it can be expressed as $Diff_i(n) = |s_i - \xi|$ in the $n$-th round . Therefore, the Data trust can be formalized as

$$DaT_A^i(n) = \begin{cases} DaT_A^i(n-1) + \frac{1-DaT_A^i(n-1)}{\rho}, Diff_i(n) < v \\ DaT_A^i(n-1) - \frac{DaT_A^i(n-1)}{\sigma}, Diff_i(n) \geq v \end{cases} \quad (1)$$

where $DaT_A^i(n-1)$ is the data trust of $i$ calculated by A in the $(n-1)$-th iteration and the $DaT_A^i(n)$ is between 0 and 1. $v$ is the most deviation value. $\rho$ and $\sigma$ is used to control the increasing and decreasing rations in $DaT_A^i(n)$. Then, the data trust of B calculated by A is represented as $DaT_A^B(t)$.

**Energy Trust.** When a CM transmits the packets to CH, the remaining energy ratio of CM, $re^t$, is added to the packets. If a CM discovers that its $re^t$ is below a certain threshold $th_{re}$, CM will be treated as an invalid or malicious node and will no longer join the normal

data transfer process. In addition, the energy consumption rate, $\Delta p = \left|p^t - p^{t-1}\right|/p^{t-1}$, is used to detect anomalies. If $\Delta p$ exceeds a certain threshold $th_{\Delta p}$, then CH considers CM to be an abnormal node and can set the energy trust of CM to 0. Therefore, the $ET_A^B(t)$ is

$$T_A^B(t) = \begin{cases} re^t(1-\Delta p), re^t > th_{re} \,\&\, \Delta \mathrm{p} < \mathrm{th}_{\Delta \mathrm{p}} \\ 0, re^t < th_{re} || \Delta p > th_{\Delta p} \end{cases} \tag{2}$$

**Forwarding Trust.** In the case of malicious behavior (i.e. the Selective Forwarding), a node may randomly drops some packets received from its neighbor nodes. As a countermeasure, node A can detect this malicious behavior through listening the forwarding activities of node B. After sending the data packets to node B, node A keeps monitoring the forwarding packets of node B. Assumed that node B forwards $p_t$ packets and deserts $q_t$ packets in a transmission round. Then, node A can evaluate the forwarding ratio as $FR_A^B(t) = p_t/(p_t + q_t)$. Then, $FT_A^B(t)$ is described as

$$T_A^B(t) = FR_A^B(t)\cos(\frac{\pi}{2} \cdot \delta_t) \tag{3}$$

**Comprehensive DT.** The comprehensive DT of node is composed of Data Trust, Energy Trust and Forwarding Trust and it may be formalized as Eq. (4).

$$\begin{gathered} DT_A^B = \omega_1 DaT_A^B + \omega_2 ET_A^B + \omega_3 FT_A^B \\ s.t.\ \min\{DaT_A^B, ET_A^B, FT_A^B\} \geq th_{DT} \end{gathered} \tag{4}$$

where $\omega_1$, $\omega_2$, and $\omega_3$ are weights of three attributes and their sum is 1 in order to limit the direct trust in [0, 1].

### 2.3 Indirect Trust(IT)

**Recommendation Trust.** The uncertainty of neighbor recommendations makes it easy to be a prime target for malicious attacks. Therefore, this paper uses the weighted Dempster-Shaffer Theory (DST) to deal with this kind of uncertainty problem [13].

Therefore, we propose the recommendation trust,

$$RC_A^B(t) = \begin{cases} 1 - \frac{\log(S_A^B(t))}{\log(\theta)}, S_A^B(t) > \theta \\ 0, Otherwise \end{cases} \tag{5}$$

to distinguish the legitimate recommendation from malicious ones where $S_A^B(t)$ is the similarity parameter between A and B. It denotes the similar extent between A and B in terms of trust. The basic theory of DST can be found in [13],

$$m_1 \oplus m_2 = \begin{cases} 0, if A = \emptyset \\ \sum_{A_i \cap A_j = A} m_1(A_i) m_2(A_j), \ , if\ \emptyset \neq A \subseteq \Omega \\ \sum_{A_i \cap A_j \neq \emptyset} m_1(A_i) m_2(A_j) \end{cases} \tag{6}$$

where $m$ is a mass function over a frame $\Omega$, $m : 2^{\Omega} \rightarrow [0, 1]$.

According to the combination rule, the recommendation trust ($RT_A^B(t)$) can be evaluated as follows. Let $U = \Omega = \{T,\text{M}, U\}$ be the identification framework where $T$, $F$ and $U$ denote the Trust, Malicious and Undecided, respectively. In D-S theory, the Mass Function (MF) is the basic factor to fuse the evidence from different sources. Let that $m_x^B$ is the MF of sensor node $x$, $m_y^B$ is the MF of sensor node $y$ and $FT_x^B$ the credibility of B at node $x$. Some weights are used to represent the credibility of referee and we represent them as $RC_*^*(t)$. If $x$ and node B are neighboring nodes and node A computes $RT_A^B(t)$ on B. The value of mass function is formalized as follows if node B has $N$ neighbor nodes.

$$\begin{cases} m_x^B(T) = \frac{RC_A^x \times RT_x^B}{\sum_{x \in N-\{A\}} RC_A^x} \\ m_x^B(M) = 0 \\ m_x^B(U) = 1 - \frac{RC_A^x \times RT_x^B}{\sum_{x \in N-\{A\}} RC_A^x} \end{cases} \tag{7}$$

where $x \in N$. Next, the different MF in above equations can be combined as

$$\begin{cases} m_x^B(T) \oplus m_y^B(F) = \frac{1}{K}\left[m_x^B(T)m_y^B(T) + m_x^B(T)m_y^B(U) + m_x^B(U)m_y^B(T)\right] \\ m_x^B(M) \oplus m_y^B(F) = \frac{1}{K}\left[m_x^B(F)m_y^B(F) + m_x^B(F)m_y^B(U) + m_x^B(U)m_y^B(F)\right] \\ m_x^B(U) \oplus m_y^B(U) = \frac{1}{K}m_x^B(U)m_y^B(U) \end{cases} \tag{8}$$

$$\begin{aligned} K = {} & m_x^B(T)m_y^B(T) + m_x^B(T)m_y^B(U) + m_x^B(U)m_y^B(U) + m_x^B(U)m_y^B(T) \\ & + m_x^B(U)m_y^B(M) + m_x^B(M)m_y^B(M) + m_x^B(M)m_y^B(U) \end{aligned} \tag{9}$$

The trust value of B is $bel(T) = m_x^B(H) \oplus m_y^B(H)$. $RC_A^B(t)$ on B is calculated by A, then.

$$T_A^B = m_x^B(T) \oplus m_y^B(T) \oplus \ldots \oplus m_y^B(T) \tag{10}$$

**Communication Trust.** The communication trust, $CT_A^B(t)$, is calculated through the packet forwarding behavior with the neighbor and it can be regarded as an index to detect the black and gray hole attack

$$T_A^B(t) = \tau FT_A^B(t) + (1 - \tau)RT_A^B(t) \tag{11}$$

In Eq. (11), the different trusts are endowed different weights according to various trusts. The parameter $\tau$ is evaluated according to the forwarding behaviors of nodes.

$$\tau = \frac{I_t(A,B)}{I_t(A,B)+M_t(A,B)} \tag{12}$$

where $I_t(A, B)$ represents that node B forwards $I_t$ packets which are received from node A. $M_t(A, B)$ denotes that node B sends $M_t$ packets except the packets of node A. Noticed that as $I_t(A, B)$ increases, $\tau$ increases too.

**IT Computation.** Two indexes, the recommendation trust and the communication trust, directly determine the IT of a node. As the DT demonstrated in Subsect. 2.2, the comprehensive IT of a node can be assessed according to above-mentioned analysis,

$$\begin{aligned} & IT_A^B = \omega_4 RT_A^B + \omega_5 CT_A^B \\ & s.t. \min\{RT_A^B, CT_A^B\} \geq th_{IT} \end{aligned} \tag{13}$$

where $\omega_4$ and $\omega_5$ represent the weights of recommendation trust and communication trust which satisfy $\sum_{i=4}^{5} \omega_i = 1$ and ensure the range of IT is 0 to 1. And $th_{IT}$ is used to determine whether these trusts are reliable or not.

### 2.4 Total Trust Computing

Through the discussion of DT and IT in Subsect. 2.2 and Subsect. 2.3, the trust of a node is related to both DT and IT.

$$T_A^B = C_A^B DT_A^B + \left(1 - C_A^B\right) IT_A^B \tag{14}$$

where $C_A^B$ shows the importance of DT in the total trust and it is quantified by

$$C_A^B = \frac{NI_A^B}{NI_A^B + n} \tag{15}$$

where $NI_A^B$ is the number of direct interactions.

### 2.5 Trust Path Selection

The goal of trust path selection is to choose the trust node as the relay node and avoid the malicious node to be selected as the next hop. In this section, we mainly consider the self-recommendation, $R_{self}$, as one of important indexes to determine whether a node has the willingness to transmit the data received from the downstream nodes.

$$R_{self} = \begin{cases} 0, S < S_{avg} \\ 1, S \geq S_{avg} \end{cases} \tag{16}$$

where $S \in [0, 1]$ denotes the current self-recommendation ratio and $S_{avg} \in [0, 1]$ represents the average self-recommendation ratio which can be assessed through collecting the self-recommendation among neighboring nodes.

We redesign the format of data packet as $\mathrm{DP} = \left\{Data, E_{res}, \mathrm{R}_{self}, \mathrm{Tt}_j, \mathrm{RT}_j^{n1}\left[RT_j^{n2}...\right]\right\}$. $E_{res}$ is the residual energy of node $j$. Node $i$ calculates $Tt_i^j$ after it receives the data packet of $j$ as shown in Algorithm 1.

**Algorithm 1.** Trust value calculation.

| | |
|---|---|
| **Inputs:** | $th_{Tt}$--threshold of total trust value,<br>$R_{self_j}$--self-recommendation of node $j$. |
| **Outputs:** | $Tt_i^j$--total trust value of node $j$. |
| 1 | Node $i$ calculates $DT_i^j$ and $IT_i^j$ of $j$ according to the TSDAM; |
| 2 | $Tt_i^j = Aggregation(DT_i^j, IT_i^j) = Tt_A^B$; |
| 3 | **If** $(Tt_i^j < th_{Tt} \& R_{self_j} == 1)$ |
| 4 | $Tt_i^j = 0.0$ and node $j$ is incredible; |
| 5 | **Else If** $(Tt_i^j >= th_{Tt} \& R_{self_j} == 0)$ |
| 6 | $Tt_i^j = th_{Tt}$ and node $j$ is credible; |
| 7 | **Else If** $(Tt_i^j >= th_{Tt} \& R_{self_j} == 1)$ |
| 8 | $Tt_i^j = Tt_i^j$ and node $j$ is credible; |
| 9 | **End If** |
| 10 | **End If** |
| 11 | **End If** |

Algorithm 1 meets two conditions. (i) The next hop node is trustable $\left(Tt_i^j >= th_{Tt}\right)$; (ii) the next hop node is willing to join in the communication with the symbol of $R_{self} = 1$. Figure 1 illustrates the process of trust selection.

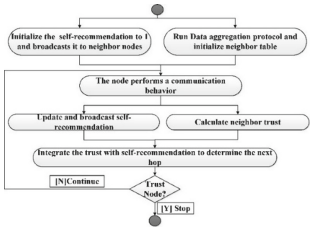



**Fig. 1.** The work flow of trust path selection

## 2.6 Secure Data Aggregation

Detailed steps for the DA process are described in Algorithm 2 in which the self-recommendation and the trust mechanism are applied.

**Algorithm 2.**The DA method of TSDAM.

| | |
|---|---|
| **Input:** | $Tt_i^j$--total trust value of node $j$ |
| **Outputs:** | the next hop and aggregation result |
| 1 | Initialize the neighbor trust and record in the trust table; |
| 2 | Nodes= {node \| the $R_{self}$ = 1 & in neighbor table}; |
| 3 | **If** (nodes≠null); |
| 4 | next-hop= {node\|$Tt_i^j$ is maximum}; |
| 5 | **Else** |
| 6 | next-hop= {node\| $Tt_i^j$ is maximum & in neighbor table}; |
| 7 | **End If** |
| 8 | Calculate $Data\_Agg(CH)$ with credible data; |
| 9 | **return** $Data\_Agg(CH)$ and next-hop node. |

Step 1: CH calculates the total trust of $s_i$ according to DT and IT discussed in Sect. 2.2 and Sect. 2.3 from the neighbors of $s_i$. Meanwhile CH judges whether $s_i$ is a trust one or not. If node $s_i$ is trustable, the $Data_i$ from $s_i$ will be aggregated. Otherwise, the CH broadcasts its *ID* to all CMs and removes the message received from the node.

Step 2: CH forwards the trusted result $Data\_Agg(CH)$ to BS through the neighbor CH.

Step 3: Node $s_i$ continuously monitors the transmission function of the neighbor. If the neighbor node receives and correctly routes a packet, the behavior is determined to be normal behavior. Otherwise, the behavior is abnormal. According to the results of the monitoring, the records in the neighbor behavior table are updated.

## 3 Experimental Simulation

We tested and verified the performance of our model and compared the performance of TSDAM with other typical trust-based DA model, such as TRPM and LDTS. The simulation parameters are demonstrated in Table 1.

**Table 1.** Simulation parameters.

| Parameter | Value |
|---|---|
| Running time | 500 s |
| Deployment region | 200 m × 200 m |
| Deployment manner of nodes | Random |
| Radio coverage | 50 m |
| Packet size | 100 bytes |
| Memory size | 50 KB |
| Battery capacity | 25 J |
| $\omega_1, \omega_2, \omega_3, \omega_4, \omega_5$ | 1/3, 1/3,1/3, 1/2, 1/2 |

### 3.1 Performance Evaluation

We set the same packet loss rate for each malicious in the case of selective forwarding and Fig. 2 shows that three models demonstrate almost the same trend with the increasing of malicious nodes. In Fig. 5, we compare the network lifetime of three schemes used in WSNs. In TSDAM, the lifetime is stable at 460 s if there are a few damaged nodes in the network. The lifetime of TSDAM decreased slightly when the number of attackers reached to 20. Meanwhile, the network lifetimes of the TRPM and LDTS models show a significant downward trend as the total number of malicious nodes increases (Fig. 3 and 4).

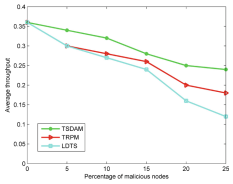



**Fig. 2.** Average throughput

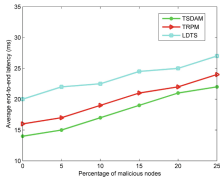



**Fig. 3.** End-to-end delay

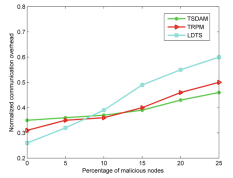



**Fig. 4.** Communication overhead

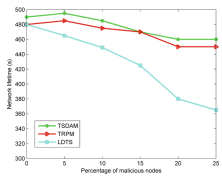



**Fig. 5.** Network lifetime

### 3.2 Trust Computation and Accuracy

Figure 6 shows that a credible node provides good service to its neighbors that results in a high value. The malicious node drops the packet continuously and reduces its trust value until the value is reduced to 0. Figure 7 shows the trust value of a normal node given by one of its neighbors while the self-recommendation mechanism is used in the network. The figure indicates that the workload can be allocated to nodes more rationally by using the mechanism.

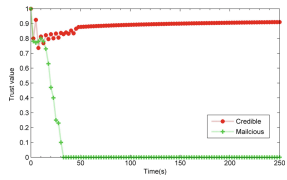



**Fig. 6.** Trust evolution

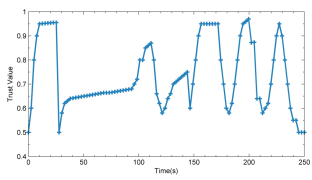



**Fig. 7.** Trust evolution with self-recommdation

## 4 Conclusion

In this paper, a trust secure data aggregation model with multiple attributes for WSNs is proposed. The model includes multiple node attributes, including forwarding behavior, the sensing data, the energy, the recommendation and the self-recommendation. Simulation experiments show that TSDAM successfully mitigates negative effect of attack on WSNs without sacrificing the network performances to a large extent. In this paper, we only demonstrate the self-recommendation and more sophisticated trust mechanisms are needed. Moreover, the trust model is sensitive to the thresholds, a more reasonable threshold mechanism should be investigated in future study.

## References

1. Rathore, H., Badarla, V., Shit, S.: Consensus-aware sociopsychological trust model for wireless sensor networks. J. Netw. Comput. Appl. **62**, 75–87 (2016)
2. She, W., Liu, Q., Tian, Z., Chen, J.S., Wang, B., Liu, W.: Blockchain trust model for malicious node detection in wireless sensor networks. IEEE Access **7**, 38947–38956 (2019)
3. Ram Prabha, V., Latha, P.: Enhanced multi-attribute trust protocol for malicious node detection in wireless sensor networks. Sādhanā **42**(2), 143–151 (2017). https://doi.org/10.1007/s12046-016-0588-2
4. Chen, Z., He, M., Liang, W., Chen, K.: Trust-aware and low energy consumption security topology protocol of wireless sensor network. J. Sens **2015**, 1–10 (2015)
5. Hu, Z., Bie, Y., Zhao, H.: Trusted tree-based trust management scheme for secure routing in wireless sensor networks. Int. J. Distrib. Sens. Netw. **11**(12), 1–13 (2015)
6. Gong, P., Chen, T.M., Xu, Q.: ETARP: an energy efficient trust-aware routing protocol for wireless sensor networks. J. Sens. **2015**, 1–10 (2015)
7. Talbi, S., Koudil, M., Bouabdallah, A., Benatchba, K.: Adaptive and dual data-communication trust scheme for clustered wireless sensor networks. Telecommun. Syst. **65**(4), 605–619 (2016). https://doi.org/10.1007/s11235-016-0254-3
8. Jan, M.A., Nanda, P.: A Sybil attack detection scheme for a forest wildfire monitoring application. Futur. Gener. Comput. Syst. **80**, 613–626 (2018)
9. Qin, D., Yang, S., Jia, S., Zhang, Y., Ma, J., Ding, Q.: Research on trust sensing based secure routing mechanism for wireless sensor network. IEEE Access **5**, 9599–9609 (2017)
10. Li, X., Zhou, F., Du, J.: LDTS: a lightweight and dependable trust system for clustered wireless sensor networks. IEEE Trans. Inf. Forensics Secur. **8**(6), 924–935 (2013)
11. Sun, B., Li, D.: A comprehensive trust-aware routing protocol with multi-attributes for WSNs. IEEE Access **6**, 4725–4741 (2017)
12. Anand, J.V.: Trust-value based wireless sensor network using compressed sensing. J. Electron. **2**(02), 88–95 (2020)
13. Busi Reddy, V., Negi, A., Venkataraman, S.: Communication and data trust for wireless sensor networks using D-S theory. IEEE Sens. J. **17**(12), 3921–3929 (2017)
14. Gu, X., Wang, J., Qiu, J.: Self-recommendation mechanism in trust calculation among nodes in WSN. Wireless Pers. Commun. **97**(3), 3705–3723 (2017)

# Lattice-Based Revocable Identity-Based Proxy Re-encryption with Re-encryption Verifiability

Xiaolei Wang(✉), Yang Wang(✉), and Mingqiang Wang(✉)

School of Mathematics, Shandong University, Jinan 250100, Shandong, China
wxl201811968@163.com, wyang1114@mail.sdu.edu.cn, wangmingqiang@sdu.edu.cn

**Abstract.** Identity-based proxy re-encryption (IB-PRE) is a type of public key cryptography that allows a proxy to convert a ciphertext under Alice's identity into another ciphertext of the same message under Bob's identity, but the proxy can not access the participants' secret keys or underlying plaintext. As far as practical application is concerned, a key revocation mechanism is an essential feature of an identity-based encryption system. By extending IB-PRE scheme, we propose a new cryptographic primitive revocable identity-based proxy re-encryption with re-encryption verifiability (RIB-VPRE), which allows the RIB-VPRE scheme to support the users' revocation, delegation of decryption rights and re-encryption verifiability at the same time. In this paper, we give the first concrete construction of collusion-resistant unidirectional RIB-VPRE on lattice, which is secure under the standard model based on learning with error (LWE) for both selective and adaptive identities.

**Keywords:** Lattice · IB-PRE · LWE

## 1 Introduction

In 2006, Green et al. [9] proposed a unidirectional identity-based proxy re-encryption (IB-PRE) scheme, which integrates the identity-based encryption (IBE) mechanism into the proxy re-encryption scheme, and uses some valid personal information as the user's public key. Due to its "identity matching public key" feature, IB-PRE effectively solves the difficult problem of public key certificate distribution and management. Since then, many attempts have been made on IB-PRE, but most of them are based on number theory problems such as DBDH. For lattice based construction, Singh et al. [16] proposed the first bidirectional IB-PRE scheme under the random oracle model, in which the re-encryption key is represented by $rk_{A\to B} = sk_A - sk_B$ where $sk_A$ and $sk_B$ is the

Supported by the National Key Research and Development Program of China (No. 2021YFA1000600), the National Key Research and Development Program of China (Grant No. 2018YFA0704702), and the National Natural Science Foundation of China (Grant No.61832012).


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 535–544, 2022.
https://doi.org/10.1007/978-3-031-19208-1_44

secret keys of the delegator Alice and the delegatee Bob. The method of obtaining the re-encryption key is derived from some EIGamal-based PRE schemes. Then, they [17] proposed a unidirectional IB-uPRE under the random oracle model, in which the secret key is composed of two trapdoors, the one is used to generate a re-encryption key and the other is used to decrypt. This scheme encrypts the message bit by bit, and the size of the re-encrypted ciphertext is larger than the original ciphertext.

The first unidirectional IB-uPRE under the standard model was proposed by Dutta et al. [7], and the secret key extraction adopts the novel trapdoor delegation technique of Micciancio and Peikert [14]. Unfortunately, it can not resist the collusion attack. In 2022, Dutta et al. [8] proposed a specific construction of collusion-resistant unidirectional IB-uPRE, which can withstand quantum attack in the Standard Model.

Most of the existing lattice-based IB-PRE schemes have implemented a security model called IND-sID-CPA security, in which a malicious third party (who is not a participant in our IB-PRE system) must declare its intended identity before seeing any public parameters. In this case, the adversary can declare his target identity after seeing the master public key and querying the re-encryption keys. For the revocable RIB-PRE scheme, we also propose a similar security model. In the IB-PRE scheme, the proxy is modeled as a semi-honest party. In practical applications, because of the interests, the proxy is prone to some dishonest transformations, we solve this problem by supporting functionality verifiability.

**Our Contributions.** Most of the existing lattice-based IB-PRE schemes only realize a weak security model called IND-sID-CPA security. We exploit the IBE scheme of katsumata et al. [12], and add algorithms such as re-encryption key generation and re-encryption to generate a unidirectional IB-PRE, for both selective and adaptive identity. For internal attacks, our scheme is collusion-resistant security. In addition, we propose a new feature called re-encryption verifiability, in which the recipient of the re-encrypted ciphertext or a third party can verify that the received ciphertext is correctly converted from the original ciphertext, thus detecting illegal activities of the proxy. We use homomorphic signature technology as a black box to realize re-encryption verifiability, and obtain the first concrete constructions of collusion-resistant unidirectional RIB-uVPRE scheme with re-encryption verifiability under the standard model based on the hardness of learning with error problem. Moreover, our scheme is decryption key exposure resistance (DKER).

## 2 Preliminaries

**Lemma 1.** *([10]) Let $q$ be a prime or some power of a prime $p$ and let $n$, $m$ be positive integers such that $m \geq 2n \log q$. Let $\sigma$ be any positive real such that $\sigma \geq \omega(\sqrt{\log n})$. Then for $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, \sigma}$, the distribution of $\mathbf{u} = \mathbf{Ae} \bmod q$ is statistically close to uniform over $\mathbb{Z}_q^n$. Furthermore, for a fixed $\mathbf{u} \in \mathbb{Z}_q^n$,*

*the conditional distribution of* $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,\sigma}$, *given* $\mathbf{Ae} \mod q = \mathbf{u}$ *for a uniformly random* $\mathbf{A}$ *in* $\mathbb{Z}_q^{n\times m}$ *is* $D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$ *with all but negligible probability.*

**Lemma 2.** *([2,14]) Let* $n$, $m$, $\bar{m}$, $q > 0$ *be positive integers with* $m \geq 2n\lceil \log q \rceil$ *and* $q$ *is a prime. Then, we have the following polynomial time algorithms*:

$\mathbf{TrapGen}(1^n, 1^m, q) \rightarrow (\mathbf{A}, \mathbf{T_A})$ : *a randomized algorithm that outputs a full rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *and a basis* $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ *for* $\Lambda_q^{\perp}(\mathbf{A})$ *such that* $\mathbf{A}$ *is statistically close to uniform and* $\|\mathbf{T_A}\|_{GS} = O(\sqrt{n \log q})$ *and* $\|\mathbf{T_A}\| = O(n \log q)$ *with overwhelming probability in* $n$.

$\mathbf{SampleLeft}(\mathbf{A}, \mathbf{F}, \mathbf{u}, \mathbf{T_A}, \sigma) \rightarrow \mathbf{e}$ : *a randomized algorithm that, given as input a full rank matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, *a matrix* $\mathbf{F} \in \mathbb{Z}_q^{n\times \bar{m}}$, *a vector* $\mathbf{u} \in \mathbb{Z}_q^n$, *a basis* $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$, *and a Gaussian parameter* $\sigma \geq \|\mathbf{T_A}\|_{GS} \cdot w(\sqrt{\log m})$, *outputs a vector* $\mathbf{e}$ *sampled from a distribution statistically close to* $D_{\Lambda_q^{\mathbf{u}}([\mathbf{A}|\mathbf{F}]),\sigma}$.

**Lemma 3.** *([6,14]) Let* $n$, $m$, $\bar{m}$, $q > 0$ *be positive integers with* $m > n$ *and* $q$ *a prime. Then, we have the following polynomial time algorithms*:

$\mathbf{ExtRndLeft}(\mathbf{A}, \mathbf{F}, \mathbf{T_A}, \sigma) \rightarrow \mathbf{T}_{[\mathbf{A}|\mathbf{F}]}$ : *a randomized algorithm that, given as input matrices* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{F} \in \mathbb{Z}_q^{n\times \bar{m}}$, *a basis* $\mathbf{T_A}$ *of* $\Lambda_q^{\perp}(\mathbf{A})$, *and a Gaussian parameter* $\sigma \geq \|\mathbf{T_A}\|_{GS} \cdot w(\sqrt{\log m})$, *outputs a matrix* $\mathbf{T}_{[\mathbf{A}|\mathbf{F}]}$ *distributed statistically close to* $(D_{\Lambda_q^{\perp}([\mathbf{A}|\mathbf{F}])})^{m+\bar{m}}$.

$\mathbf{ExtRndRight}(\mathbf{A}, \mathbf{G}, \mathbf{R}, \mathbf{T_G}, \sigma) \rightarrow \mathbf{T}_{[\mathbf{A}|\mathbf{AR}+\mathbf{G}]}$ : *a randomized algorithm that, given as input full rank matrices* $\mathbf{A}, \mathbf{G} \in \mathbb{Z}_q^{n\times m}$, *a matrix* $\mathbf{R} \in \mathbb{Z}^{m\times m}$, *a basis* $\mathbf{T_G}$ *of* $\Lambda_q^{\perp}(\mathbf{G})$, *and a Gaussian parameter* $\sigma \geq \|\mathbf{R}\|_2 \|\mathbf{T_G}\|_2 \cdot w(\sqrt{\log n})$ *outputs a matrix* $\mathbf{T}_{[\mathbf{A}|\mathbf{AR}+\mathbf{G}]}$ *distributed statistically close to* $(D_{\Lambda_q^{\perp}(\mathbf{T}_{[\mathbf{A}|\mathbf{AR}+\mathbf{G}]}),\sigma})^{2m}$.

We recall some useful facts that will be used in our paper.

**Lemma 4 (Leftover Hash Lemma).** *Let* $n$, $m$, $k$ *be positive integers,* $q \geq 2$ *is a prime. Assume further that* $m > (n+1)\log q + w(\log n)$, $k$ *is polynomial in* $n$, $\mathbf{R} \leftarrow \{-1, 1\}^{m\times k}$. *Let matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *sampled uniformly in* $\mathbb{Z}_q^{n\times m}$ *and* $\mathbb{Z}_q^{n\times k}$, *respectively. Then the distribution of the pair* $(\mathbf{A}, \mathbf{AR})$ *is negligibly close in* $n$ *to the distribution of* $(\mathbf{A}, \mathbf{B})$.

**Lemma 5 (Noise Re-randomization, [13]).** *Let* $q$, $l$, $m$ *be positive integers and* $r$ *a positive real satisfying* $r > \max\{\omega(\sqrt{\log m}), \omega(\sqrt{\log l})\}$. *Let* $\mathbf{b} \in \mathbb{Z}_q^m$ *be arbitrary and* $\mathbf{z}$ *chosen from* $D_{\mathbb{Z}^m,r}$. *Then for any* $\mathbf{V} \in \mathbb{Z}^{m\times l}$ *and positive real* $\sigma > s_1(\mathbf{V})$, *there exists a* PPT algorithm $\mathbf{ReRand}(\mathbf{V}, \mathbf{b} + \mathbf{z}, r, \sigma)$ that outputs $\mathbf{b'}^T = \mathbf{b}^T\mathbf{V} + \mathbf{z'}^T \in \mathbb{Z}_q^l$ where $\mathbf{z'}$ is distributed statistically close to $D_{\mathbb{Z}^l,2r\sigma}$.

**Definition 1 (*Bits* and *Power*2 functions).** *Let* $\mathbf{a}$, $\mathbf{b} \in \mathbb{Z}_q^n$ *and* $k = \lceil \log_2 q \rceil$. *Let* $Bits(\mathbf{a}) = (a_{1,0}, \cdots, a_{1,k-1}, \cdots, a_{n,0}, \cdots, a_{n,k-1})$, *where* $a_{i,j}$ *is the* $j$*-th bit in* $a_i$*'s binary representation, bits ordered least significant to most significant. Let* $Power2(\mathbf{b}) = (b_1, \cdots, 2^{k-1}b_1, \cdots, b_n, \cdots, 2^{k-1}b_n)$, *a* $nk$*-dimensional vector.*

**Definition 2 (FRD map, [2]).** *Let* $q$ *be a prime and* $n$ *a positive integer. We say that a function* $H : \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q^{n\times n}$ *is an encoding with full-rank differences* (FRD) *map if:* $\forall$ *distinct* $id_1$, $id_2 \in \mathbb{Z}_q^n$, *the matrix* $H(id_1) - H(id_2) \in \mathbb{Z}_q^{n\times n}$ *is full rank;* $\forall$ $id \in \mathbb{Z}_q^n \backslash \{\mathbf{0}\}$, *the matrix* $H(id) \in \mathbb{Z}_q^{n\times n}$ *is full rank;* $H$ *is computable in polynomail time in* $n \log q$.

**The Binary-Tree Data Structure.** We use $BT$ to represent a binary-tree. If *root* represents a root node and $v$ represents a leaf node, then Path($v$) represents the set of all nodes (both $v$ and *root* inclusive) on the path from $v$ to *root*. If $\theta$ represents a non-leaf node, then $\theta_l$, $\theta_r$ represent the left and right child of $\theta$, respectively. Here we present an KuNodes algorithm, a binary-tree $BT$, a revocation list $RL$ and a time $t$ as input and a set $Y$ as output. At time $t$, the KGC determines the minimal set $Y$ of nodes in $BT$ such that none of the nodes in $RL$ with corresponding time $\leq t$ have any ancestor (or, themselves) in the set $Y$, and all other leaf nodes have exactly one ancestor (or, themselves) in the set. The KuNodes algorithm is in the full version.

The binary-tree $BT$ is maintained by the KGC, and each user is assigned to a leaf node $v$. The KGC provides each user with a set of private keys, which is composed of all nodes in Path($v$). At time $t$, the KGC issues a key update for all nodes in set $Y$. If the set $Y$ and Path($v$) have a common node, the user $id$ assigned to leaf node $v$ can generate a legitimate decryption key within time $t$.

### 2.1 Definition and Security Model of RIB-VPRE

Definition of homomorphic signature scheme in [11].

**Definition 3 (Syntax of RIB-uVPRE).** *A unidirectional revocable identity-based proxy re-encryption* (RIB-uVPRE) *scheme with re-encryption verifiability as follows:*
**Setup**$(1^\lambda, N) \rightarrow (PP, msk, RL, ST)$ : *Outputs a public parameter $PP$ and a master secret key $msk$, a revocation list $RL$, and a state $ST$.*
**PriKeyGen**$(PP, msk, id, ST) \rightarrow (sk_{id}, ST)$ : *Outputs a private key $sk_{id}$ corresponding to the identity $id$ and an updated state $ST$.*
**KeyUpd**$(PP, msk, ST, RL, t) \rightarrow ku_t$ : *Outputs a key update $ku_t$.*
**DecKeyGen**$(sk_{id}, ku_t) \rightarrow dk_{id,t}$ : *Outputs a decryption key $dk_{id,t}$ or a special symbol $\perp$ indicating that $id$ was revoked.*
**Encrypt**$(PP, id, t, m) \rightarrow ct_{id,t}$ : *Outputs a ciphertext $ct_{id,t}$.*
**ReKeyGen**$(PP, id_i, id_j, dk_{id_i,t}, t) \rightarrow (rk_{i\rightarrow j,t}, vk_{i\rightarrow j,t})$ : *Outputs a re-encryption key $rk_{i\rightarrow j,t}$ and a re-encryption verification key $vk_{i\rightarrow j,t}$.*
**ReEncrypt**$(PP, ct_{id_i,t}, rk_{i\rightarrow j,t}) \rightarrow ct_{id_j,t}$ : *Outputs a re-encryption ciphertext $ct_{id_j,t}$ under the identity $id_j$.*
**ReEncVer**$(PP, ct_{id_i,t}, ct_{id_j,t}, vk_{i\rightarrow j,t}) \rightarrow 1/\perp$: *Outputs 1 if the re-encrypted ciphertext $ct_{id_j,t}$ is correctly transformed from the original ciphertext or $\perp$.*
**Decrypt**$(PP, ct_{id,t}, dk_{id,t}) \rightarrow m$ : *Outputs a plaintext $m$ or a error symbol $\perp$.*
**KeyRev**$(id, t, RL, ST) \rightarrow RL$ : *Outputs an updated revocation list $RL$.*

A single-hop unidirectional RIB-uVPRE is correct, then the following two properties hold: Decryption correctness. for any $m \in \mathcal{M}$, $id_1, id_2 \in \mathcal{I}$, $t \in \mathcal{T}$, we have **Decrypt**$(PP, ct_{id_i,t}, dk_{id_i,t}) = m$, **Decrypt**$(PP, dk_{id_j,t}, ct_{id_j,t} \leftarrow$ **ReEncrypt** $(PP, ct_{id_i,t}, rk_{i\rightarrow j,t})) = m$. Verification correctness. This scheme satisfies verification correctness if for all $ct_{id_j}$ generated by **ReEncrypt** $(PP, ct_{id_i,t}, rk_{i\rightarrow j,t})$ with the re-encryption key $rk_{i\rightarrow j,t} \leftarrow$ **ReKeyGen**$(PP, id_i,$

$id_j, dk_{id_i,t}$), we have that the probability $\Pr[\mathbf{ReEncVer}(PP, ct_{id_i,t}, ct_{id_j,t}, vk_{i\to j,t}) = 1] = 1$.

In previous works, this scheme had no complete definition of security. Combined with the security of proxy re-encryption scheme and key revocation mechanism, we give a perfect security definition for RIB-VPRE in our work. The security model is described by the game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The adversary $\mathcal{A}$ is given the ability to access the decryption key oracle $\mathcal{O}^{\mathbf{DecKeyGen}(\cdot)}$, so our scheme is decryption key exposure resistance(DKER).

**Definition 4 (Single-hop RIB-uVPRE IND-sID-CPA security).** *To describe the security model, we first classify all users into honest (HU) and corrupted (CU). According to the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$, the indistinguishability of plaintext under adaptive chosen-plaintext and selective chosen-identity attack* (IND-sID-CPA) *of a* RIB-uVPRE *scheme is defined.*

*Initial. The adversary $\mathcal{A}$ first outputs the challenge identity $id^*(\in HU)$ and time $t^*$, and also some information state it wants to preserve.*

*Setup. The challenger $\mathcal{C}$ performs* $\mathbf{Setup}(1^\lambda, N)$ *to get* $(PP, msk, RL, ST)$ *and gives $PP$ to adversary $\mathcal{A}$.*

*Query phase 1. $\mathcal{A}$ makes the following queries.* $\mathcal{O}^{\mathbf{Pri}(\cdot)}$ *: On input a user identity $id \in CU$, $\mathcal{C}$ returns $sk_{id}$ by running* $\mathbf{PriKeyGen}(PP, msk, id, ST)$*. Otherwise, return $\perp$.*

$\mathcal{O}^{\mathbf{Upd}(\cdot)}$ *: On input a time period $t$, $\mathcal{C}$ returns $ku_t$ by running* $\mathbf{KeyUpd}(PP, msk, t, RL, ST)$*.*

$\mathcal{O}^{\mathbf{DecKey}(\cdot)}$ *: On input a user identity $id \in CU$ and a time $t$, if $id \notin RL$, $\mathcal{C}$ returns $dk_{id,t}$ by running* $\mathbf{DecKey}(sk_{id}, ku_t)$*. Otherwise, return $\perp$.*

$\mathcal{O}^{\mathbf{ReKey}(\cdot)}$ *: On input two identities $id_i, id_j \notin RL$, a time $t$, if $id_i, id_j \in HU$ or $id_i, id_j \in CU$ or $id_i \in CU, id_j \in HU$, $\mathcal{C}$ returns $rk_{i\to j,t}$ by running* $\mathbf{ReKeyGen}(PP, id_i, id_j, dk_{id_i,t}, t)$*. Otherwise, return $\perp$.*

$\mathcal{O}^{\mathbf{ReEnc}(\cdot)}$ *: On input two identities $id_i, id_j \notin RL$, and a cipertext $ct_{id_i,t}$, if $id_i, id_j \in HU$ or $id_i, id_j \in CU$ or $id_i \in CU, id_j \in HU$, $\mathcal{C}$ returns $ct_{id_j,t}$ by running* $\mathbf{ReEncrypt}(PP, ct_{id_i,t}, rk_{i\to j,t})$*. Otherwise, return $\perp$.*

$\mathcal{O}^{\mathbf{ReEncVer}(\cdot)}$ *: On input cipertexts $ct_{id_i,t}$ and $ct_{id_j,t}$, $\mathcal{C}$ returns 1 or 0 by running* $\mathbf{ReEncVer}(PP, ct_{id_i,t}, ct_{id_j,t}, vk_{i\to j,t})$*. Otherwise, return $\perp$.*

$\mathcal{O}^{\mathbf{Revoke}(\cdot)}$ *: On input identity $id$ and $t$, $\mathcal{C}$ returns an updated recocation list $RL$ by running* $\mathbf{Revoke}(id, t, RL, ST)$*.*

*Challenge phase. $\mathcal{A}$ presents the same length challenge messages $(m_0, m_1) \in \mathcal{M}$. The challenger picks a random bit $b \in \{0, 1\}$, and returns the challenge ciphertext* $ct_{id^*,t^*} \leftarrow \mathbf{Encrypt}(PP, id^*, t^*, m_b)$*.*

*Query phase 2. $\mathcal{A}$ continues making queries as in Query phase 1.*

*Guess. $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ and wins this game if $b = b'$. The advantage of adversary $\mathcal{A}$ is defined as* $Adv_{\mathcal{A}}^{\text{IND-sID-CPA}}(\lambda, id^*) = \left|Pr[b = b'] - \frac{1}{2}\right|$*.*

*The following restrictions must always hold:*

$\mathcal{O}^{\mathbf{Upd}(\cdot)}$ *and* $\mathcal{O}^{\mathbf{KeyRev}(\cdot)}$ *can be queried on time which is greater than or equal to the time of all previous queries, i.e., the adversary is allowed to query only in*

*non-decreasing order of time. Also, the* $\mathcal{O}^{\mathbf{KeyRev}(\cdot)}$ *cannot be queried at time* $t$ *if* $\mathcal{O}^{\mathbf{Upd}(\cdot)}$ *was queried on* $t$.
*Then, we say an* RIB-uVPRE *scheme is* IND-sID-CPA *secure if the function* $Adv_{\mathcal{A}}^{\text{IND-sID-CPA}}$ *is negligible for any* PPT *adversary* $\mathcal{A}$ *whose time complexity is polynomial in* $\lambda$. *If adversary does not announce the challenge identity and time period* $(id^*, t^*)$ *at the beginning of the game, but at the challenge phase* (*The only limitation is that there was no previous key query about challenging identity.*), *the resulting security concept is expressed as* IND-ID-CPA.

## 3 Selective-RIB-VPRE Scheme

Our RIB-uVPRE scheme is described as follows.
$\mathbf{SetUp}(1^n, N)$ : On input a parameter $n$ and a maximal number $N$ of users. Use the $\mathbf{Trapdoor}(1^n, 1^m, q)$ algorithm to select uniformly random matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ with basis $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$ and $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times m}$ with basis $\mathbf{T}_{\bar{\mathbf{A}}} \in \mathbb{Z}^{m\times m}$. Select two uniformly random matrices $\mathbf{B}_1, \mathbf{B}_2 \leftarrow \mathbb{Z}_q^{n\times m}$. Select a uniformly random vector $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. Select a FRD map $H(\cdot)$ as in definition 2. Let $RL$ be an empty set and $BT$ be a binary-tree with at least $N$ leaf nodes, which denote $N$ children users. Set a state $ST := BT$. Output $RL$, $ST$, $PP := \{\mathbf{A}, \bar{\mathbf{A}}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{u}, H\}$, $msk := \{\mathbf{T_A}, \mathbf{T}_{\bar{\mathbf{A}}}\}$.
$\mathbf{PriKeyGen}(PP, msk, ST, id)$ : On input the public parameter $PP$, the master key $msk$, the state $ST$ and an identity $id \in \mathbb{Z}_q^n$. It picks an unassigned leaf node $v_{id}$ from $ST$ and stores $id \in \mathbb{Z}_q^n$ in that node. For any $\theta \in \text{Path}(BT, v_{id})$, if $\mathbf{u}_{\theta,id}$, $\mathbf{u}_{\theta,t}$ are undefined, then $\mathbf{u}_{\theta,id} \leftarrow \mathbb{Z}_q^n$, $\mathbf{u}_{\theta,t} := \mathbf{u} - \mathbf{u}_{\theta,id}$. Store $\mathbf{u}_{\theta,id}$, $\mathbf{u}_{\theta,t}$ in node $\theta$ and update a state $ST$. Sample $\mathbf{e}_{\theta,id} \leftarrow \mathbf{SampleLeft}(\mathbf{A}, \mathbf{B}_1 + H(id)\mathbf{G}, \mathbf{T_A}, \mathbf{u}_{\theta,id}, \sigma)$ for $\theta \in \text{Path}(BT, v_{id})$. Run $\mathbf{T}_{[\bar{\mathbf{A}}|\mathbf{B}_1+H(id)\mathbf{G}]} \leftarrow \mathbf{ExtRndLeft}\ (\bar{\mathbf{A}}, \mathbf{B}_1+H(id)\mathbf{G}, \mathbf{T}_{\bar{\mathbf{A}}}, \bar{\sigma})$. Output a private key $sk_{id} := (\{(\theta, \mathbf{e}_{\theta,id})\}_{\theta\in\text{path}(v_{id})}, \mathbf{T}_{[\bar{\mathbf{A}}|\mathbf{B}_1+H(id)\mathbf{G}]})$ and a state $ST$.
$\mathbf{KeyUpd}(PP, msk, ST, RL, t)$ : On input the public parameter $PP$, the master key $msk$, the state $ST$, the revocation list $RL$ and a time $t \in \mathbb{Z}_q^n$. For any $\theta \in \text{KUNodes}(BT, RL, t)$, if $\mathbf{u}_{\theta,id}$, $\mathbf{u}_{\theta,t}$ are undefined, then $\mathbf{u}_{\theta,t} \leftarrow \mathbb{Z}_q^n$, $\mathbf{u}_{\theta,id} := \mathbf{u} - \mathbf{u}_{\theta,t}$. Store $\mathbf{u}_{\theta,id}$, $\mathbf{u}_{\theta,t}$ in node $\theta$. Sample $\mathbf{e}_{\theta,t} \leftarrow \mathbf{SampleLeft}(\mathbf{A}, \mathbf{B}_2 + H(t)\mathbf{G}, \mathbf{T_A}, \mathbf{u}_{\theta,t}, \sigma)$ for $\theta \in \text{KUNodes}(BT, RL, t)$. Here $\mathbf{e}_{\theta,t} \in \mathbb{Z}^{2m}$ satisfies $[\mathbf{A}|\mathbf{B}_2 + H(t)\mathbf{G}]\mathbf{e}_{\theta,t} = \mathbf{u}_{\theta,t}$. Output a key update $ku_t := \{(\theta, \mathbf{e}_{\theta,t})\}_{\theta\in\text{KUNodes}(BT,RL,t)}$. $\mathbf{DecKeyGen}(sk_{id}, ku_t)$ : On input a private key $sk_{id}$ and a key update $ku_t$. Extract $P = \text{Path}(BT, v_{id})$ in $sk_{id}$, and $K = \text{KUNodes}(BT, RL, t)$ in $ku_t$. If $P \cap K = \emptyset$, output $\perp$. Otherwise for the unique node $\theta^* \in P \cap K$, set $\mathbf{dk}_{id,t} = [\mathbf{e}^L_{\theta^*,id} + \mathbf{e}^L_{\theta^*,t} | \mathbf{e}^R_{\theta^*,id} | \mathbf{e}^R_{\theta^*,t}]$. Run $\bar{\mathbf{dk}}_{id,t} \leftarrow \mathbf{SampleLeft}([\bar{\mathbf{A}}|\mathbf{B}_1 + H(id)\mathbf{G}], \mathbf{B}_2 + H(t)\mathbf{G}, \mathbf{T}_{[\bar{\mathbf{A}}|\mathbf{B}_1+H(id)\mathbf{G}]}, \mathbf{u}, \sigma)$. Output a decryption key $dk_{id,t} := (\mathbf{dk}_{id,t}, \bar{\mathbf{dk}}_{id,t})$. $\mathbf{Encrypt}(PP, id, t, m)$ : On input the parameter $PP$, an identity $id \in \mathbb{Z}_q^n$, a time $t \in \mathbb{Z}_q^n$ and a message $m$. Construct $\mathbf{A}_{id,t} := [\mathbf{A}|\mathbf{B}_1 + H(id)\mathbf{G}|\mathbf{B}_2 + H(t)\mathbf{G}]$ and $\bar{\mathbf{A}}_{id,t} := [\bar{\mathbf{A}}|\mathbf{B}_1 + H(id)\mathbf{G}|\mathbf{B}_2 + H(t)\mathbf{G}]$. Select uniformly random vectors $\mathbf{s}, \bar{\mathbf{s}} \xleftarrow{s} \mathbb{Z}_q^n$. Sample error vectors $e_0 \leftarrow D_{\mathbb{Z},\alpha q}$, $\mathbf{e}_1, \bar{\mathbf{e}}_1 \leftarrow D_{\mathbb{Z}^{3m},\alpha' q}$ and set $c_0 = \mathbf{u}^T(\mathbf{s} +$

$\bar{\mathbf{s}}) + e_0 + m\lfloor \frac{q}{2} \rfloor, \mathbf{c}_1 = \mathbf{A}_{id,t}^T \mathbf{s} + \mathbf{e}_1, \bar{\mathbf{c}}_1 = \bar{\mathbf{A}}_{id,t}^T \bar{\mathbf{s}} + \bar{\mathbf{e}}_1$. Output a ciphertext $ct_{id,t} := (c_0, \mathbf{c}_1, \bar{\mathbf{c}}_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^{3m}$.

**ReKeyGen**$(PP, dk_{id_i,t}, id_i, id_j)$ : On input the parameter $PP$, the decryption key of $i$-th user $dk_{id_i,t} = (\mathbf{dk}_{id_i,t}, \bar{\mathbf{dk}}_{id_i,t})$ and the identity of $j$-th user $id_j$. Construct $\mathbf{A}_{id_j,t}$, $\bar{\mathbf{A}}_{id_j,t}$. Select $\mathbf{r}_1, \mathbf{r}_3 \leftarrow D_{\mathbb{Z},r}^{3mk \times n}$ and $\mathbf{r}_2, \mathbf{r}_4 \leftarrow D_{\mathbb{Z},r}^{3mk \times 1}$. Compute the re-encryption key $rk_{i \to j,t} := \begin{bmatrix} 1 & \mathbf{0}_{1\times 3m} & \mathbf{0}_{1\times 3m} \\ \mathbf{r}_1\mathbf{u} + \mathbf{r}_2 - Power2(\mathbf{dk}_{id_i,t}) & \mathbf{r}_1\mathbf{A}_{id_j,t} & \mathbf{0}_{3mk\times 3m} \\ \mathbf{r}_3\mathbf{u} + \mathbf{r}_4 - Power2(\bar{\mathbf{dk}}_{id_i,t}) & \mathbf{0}_{3mk\times 3m} & \mathbf{r}_3\bar{\mathbf{A}}_{id_j,t} \end{bmatrix}$.

Generate HS key $(hssk, hsvk) = \mathbf{HS.KeyGen}(n, 6mk+1)$. Parse each row from $rk_{i\to j,t}$ as $w_i \in \mathbb{Z}_q^{6m+1}$ $(1 \le i \le 6mk+1)$, then sign each $w_i$ using the algorithm $\sigma_i = \mathbf{HS.Sign}(hssk, w_i)$. Set $vk_{i\to j,t} := \{\sigma_i\}_{i\in[6mk+1]}$. Output a proxy re-encryption key $rk_{i\to j,t}$ and a re-encryption verification key $vk_{i\to j,t} = \{\sigma_i\}_{i\in[6mk+1]}$. **ReEncrypt**$(PP, ct_{id_i,t}, rk_{i\to j,t})$ : On input the ciphertext of $i$-th user $ct_{id_i,t} = (c_{i0}, \mathbf{c}_{i1}, \bar{\mathbf{c}}_{i1})$ and the re-encryption key $rk_{i\to j,t}$. Perform the following steps: Compute $ct_{id_j,t}^T = [c_{i0}|Bits(\mathbf{c}_{i1})^T|Bits(\bar{\mathbf{c}}_{i1})^T] \cdot rk_{i\to j,t}$. Compute a signature $\sigma_{i\to j} = \mathbf{HS.SignEval}(g, \sigma_i (1 \le i \le 6mk+1))$ homomorphically where the circuit $g(rk_{i\to j,t})$ is defined by original ciphertext $ct_{id_i,t} = (c_{i0}, \mathbf{c}_{i1}, \bar{\mathbf{c}}_{i1})$ as follows: $g(rk_{i\to j,t}) = [c_{i0}|Bits(\mathbf{c}_{i1})^T|Bits(\bar{\mathbf{c}}_{i1})^T] \cdot rk_{i\to j,t}$. Output a proxy re-encryption ciphertext $ct_{id_j,t}$ and a signature $\sigma_{i\to j}$.

**ReEncVer**$(hsvk, g, ct_{id_j,t}, \sigma_{i\to j})$ : On input the verification key $hsvk$, a circuit $g$, a ciphertext $ct_{id_j,t}$, and a signature $\sigma_{i\to j}$. Run $0/1 \leftarrow \mathbf{HS.Verify}(hsvk, g, ct_{id_j,t}, \sigma_{i\to j})$, where the circuit $g(rk_{i\to j,t}) = ct_{id_j,t}^T$. Output $0/1$.

**Decrypt**$(ct_{id,t}, dk_{id,t})$ : On input a ciphertext $ct_{id,t} = (c_0, \mathbf{c}_1, \bar{\mathbf{c}}_1)$ and the user's decryption key $dk_{id,t}$. Compute $m' = (c_0, \mathbf{c}_1^T, \bar{\mathbf{c}}_1^T)[1, -\mathbf{dk}_{id,t}, -\bar{\mathbf{dk}}_{id,t}]^T \in \mathbb{Z}_q$. Output 0 if $m'$ is closer to 0 than to $\lfloor q/2 \rfloor \mod q$; Otherwise output 1.

### 3.1 Correctness and Security Analysis

In this part, we prove the correctness and parameter selection are in the full version.

**Theorem 1.** *The above scheme is* IND-sID-CPA *secure assuming the hardness of decision-*$\mathrm{LWE}_{m,n,q,\chi}$.

*Proof.* We show that a PPT adversary cannot distinguish between the games.
**Game 0** : This is the original IND-sID-CPA game from definition 4.
**Game 1** : Let $id^*$ be the identity and $t^*$ be the time that $\mathcal{A}$ intends to attack. The Game 1 challenger randomly selects $\mathbf{R}_i^* \in \{-1, 1\}^{m\times m}$ for $i \in [2]$ and constructs $\mathbf{B}_1 := \bar{\mathbf{A}}\mathbf{R}_1^* - H(id^*)\mathbf{G}$, $\mathbf{B}_2 := \bar{\mathbf{A}}\mathbf{R}_2^* - H(t^*)\mathbf{G}$. The challenger samples $\bar{\mathbf{dk}}_{id^*,t^*} \leftarrow D_{\mathbb{Z}^{3m},\sigma}$ and sets $[\bar{\mathbf{A}}|\bar{\mathbf{A}}\mathbf{R}_1^*|\bar{\mathbf{A}}\mathbf{R}_2^*]\bar{\mathbf{dk}}_{id^*,t^*} = \mathbf{u}$. The remainder of the game is unchanged. The challenger maintains that matrices $\mathbf{R}_1^*$, $\mathbf{R}_2^*$ and vector $\bar{\mathbf{dk}}_{id^*,t^*}$ are part of $msk$. In the challenge phase, the challenger uses $\mathbf{R}_1^*$ and $\mathbf{R}_2^*$ as random matrices to construct the challenge ciphertext.

Due to the leftover hash lemma 5, $(\bar{\mathbf{A}}, \bar{\mathbf{A}}\mathbf{R}_i^*)$ for $i \in [2]$ is statistically indistinguishable with uniform distribution. Hence, $(\bar{\mathbf{A}}, \bar{\mathbf{A}}\mathbf{R}_1^* - H(id^*)\mathbf{G}, \bar{\mathbf{A}}\mathbf{R}_2^* -$

$H(t^*)\mathbf{G}, [\bar{\mathbf{A}}|\bar{\mathbf{A}}\mathbf{R}_1^*|\bar{\mathbf{A}}\mathbf{R}_2^*]\bar{\mathbf{dk}}_{id^*,t^*})$ is statistically indistinguishable with uniform distribution.

**Game 2** : In this game, we generate $\bar{\mathbf{A}}$ as a random matrix in $\mathbb{Z}_q^{n\times m}$ instead of generating it with a trapdoor. The challenger runs **ExtRndRight** to create a trapdoor $\mathbf{T}_{[\bar{\mathbf{A}}|\bar{\mathbf{A}}\mathbf{R}_1^*+(H(id)-H(id^*))\mathbf{G}]}$. We have, to answer a secret key query against $id \in CU$, the challenger will construction $\mathbf{A}_{id} = [\mathbf{A}|\mathbf{A}\mathbf{R}_1^* + (H(id) - H(id^*))\mathbf{G}]$ and $\bar{\mathbf{A}}_{id} = [\bar{\mathbf{A}}|\bar{\mathbf{A}}\mathbf{R}_1^* + (H(id) - H(id^*))\mathbf{G}]$. The challenger uses $\mathbf{T}_{\mathbf{A}}$ and $\mathbf{T}_{\mathbf{G}}$ to generate a private key query $sk_{id} = (\{(\theta, \mathbf{e}_{\theta,id})\}_{\theta\in \mathrm{path}(v_{id})}, \mathbf{T}_{[\bar{\mathbf{A}}|\bar{\mathbf{A}}\mathbf{R}_1^*+(H(id)-H(id^*))\mathbf{G}]})$ and sends $sk_{id}$ to the adversary $\mathcal{A}$. The challenger $\mathcal{C}$ will send $\perp$, against the secret key query for $id \in HU$. The challenger $\mathcal{C}$ uses $\mathbf{T}_{\mathbf{A}}$ to answer a updated key query against $t$. To answer a decryption key query against $id \in CU$ and a time $t$, the challenger will construction $\mathbf{A}_{id,t} = [\mathbf{A}|\mathbf{A}\mathbf{R}_1^* + (H(id) - H(id^*))\mathbf{G}|\mathbf{A}\mathbf{R}_2^* + (H(t) - H(t^*))\mathbf{G}]$ and $\bar{\mathbf{A}}_{id,t} = [\bar{\mathbf{A}}|\bar{\mathbf{A}}\mathbf{R}_1^* + (H(id) - H(id^*))\mathbf{G}|\bar{\mathbf{A}}\mathbf{R}_2^* + (H(t) - H(t^*))\mathbf{G}]$. The challenger $\mathcal{C}$ uses $\mathbf{T}_{\mathbf{A}}$ and $\mathbf{T}_{\mathbf{G}}$ to generate a decryption key query $dk_{id,t}$. For the re-encryption key query from $id_i = id^*$ to $id_j \in HU$ in time $t = t^*$, the challenger will compute $\mathbf{A}_{id_j,t^*}$ and $\bar{\mathbf{A}}_{id_j,t^*}$, then $rk_{i^*\to j,t^*} := \begin{bmatrix} 1 & \mathbf{0}_{1\times 3m} & \mathbf{0}_{1\times 3m} \\ \mathbf{r}_1\mathbf{u} + \mathbf{r}_2 - Power2(\mathbf{dk}_{id^*,t^*}) & \mathbf{r}_1\mathbf{A}_{id_j,t^*} & \mathbf{0}_{3mk\times 3m} \\ \mathbf{r}_3\mathbf{u} + \mathbf{r}_4 - Power2(\bar{\mathbf{dk}}_{id^*,t^*}) & \mathbf{0}_{3mk\times 3m} & \mathbf{r}_3\bar{\mathbf{A}}_{id_j,t^*} \end{bmatrix}$. For other re-encryption key queries the challenger maintains the restrictions as in definition 4 and computes $rk_{i\to j,t}$ according to the algorithm **ReKeyGen** to reply the adversary. For re-encryption query challenger maintain the restrictions as in definition 4 and computes $\mathbf{ReEnc}(rk_{i\to j}, ct_{i\to j,t})$ according to the algorithm **ReEnc** to reply the adversary.

Due to the property of gadget matrix $\mathbf{G}$ and function $H(\cdot)$, we know a trapdoor $\mathbf{T}_{\mathbf{G}}$ which is also a trapdoor for $(H(id) - H(id^*))\mathbf{G}$ if $id \neq id^*$. Due to Lemma 2 and 3, since the sampled vectors and the extended trapdoors are statistically independent from the trapdoors provided as input, this makes a negligible difference.

**Game 3** : The challenger samples $e_0 \leftarrow D_{\mathbb{Z},\alpha q}$, $\mathbf{e}_1 \leftarrow D_{\mathbb{Z}^{3m},\alpha' q}$, and $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,\alpha q}$. It computes $v = \mathbf{u}^T\bar{\mathbf{s}}+e_0, \mathbf{v} = \bar{\mathbf{A}}^T\bar{\mathbf{s}}+\mathbf{e}$, then $c_0^* = v+\mathbf{u}^T\mathbf{s}+m_b\lfloor\frac{q}{2}\rfloor$, $\mathbf{c}_1^* = \mathbf{A}_{id^*,t^*}^T\mathbf{s}+\mathbf{e}_1$, where $b$ is the random bit chosen by the challenger. It sets $\mathbf{R}^* = [\mathbf{R}_1^*|\mathbf{R}_2^*]$ and runs $\mathbf{ReRand}([\mathbf{I}_m|\mathbf{R}^*], \mathbf{v}, \alpha q, \alpha'/2\alpha) \to \bar{\mathbf{c}}_1^*$. Where, $\mathbf{I}_m$ is the identity matrix. Finally, it outputs the challenge ciphertext as follows: $ct^* = (c_0^*, \mathbf{c}_1^*, \bar{\mathbf{c}}_1^*)$.

Due to the noise re-randomization lemma 5, we have $(\bar{\mathbf{c}}_1^*)^T = (\bar{\mathbf{A}}^T\bar{\mathbf{s}})^T[\mathbf{I}_m|\mathbf{R}^*] + \bar{\mathbf{e}}_1^T = \bar{\mathbf{s}}^T[\bar{\mathbf{A}}|\mathbf{B}_1 + H(id^*)\mathbf{G}|\mathbf{B}_1 + H(id^*)\mathbf{G}] + \bar{\mathbf{e}}_1^T$, where $\bar{\mathbf{e}}_1$ is distributed statistically close to $D_{\mathbb{Z}^{3m},\alpha' q}$.

**Game 4** : The challenger samples $w \leftarrow \mathbb{Z}_q$, $e_0 \leftarrow D_{\mathbb{Z},\alpha q}$, $\mathbf{w} \leftarrow \mathbb{Z}_q^m$, $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,\alpha q}$ to calculate $v = w + e_0 \in \mathbb{Z}_q$, $\mathbf{v} = \mathbf{w} + \mathbf{e}$ and runs the algorithm **ReRand** as in Game 3. Finally, it outputs $ct^* = (c_0^*, \mathbf{c}_1^*, \bar{\mathbf{c}}_1^*)$.

We now show that Game 3 is statistically indistinguishable from Game 4. The proof process is in the full version.

## 4 Adaptive-RIB-VPRE Scheme

Our RIB-uVPRE scheme is described as follows.

**SetUp**$(1^n, N)$ : On input a security parameter $n$ and a maximal number $N$ of users. Use the **Trapdoor**$(1^n, 1^m, q)$ algorithm to select uniformly random matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times m}$ with basis $\mathbf{T_A} \in \mathbb{Z}^{m\times m}$, $\mathbf{T}_{\bar{\mathbf{A}}} \in \mathbb{Z}^{m\times m}$. Select $\mathbf{B}_1, \cdots, \mathbf{B}_l \leftarrow \mathbb{Z}_q^{n\times m}$ and $\mathbf{D}_1, \cdots, \mathbf{D}_l \leftarrow \mathbb{Z}_q^{n\times m}$. Select a uniformly random vector $\mathbf{u} \leftarrow \mathbb{Z}_q^n$. Let $RL$ be an empty set and $BT$ be a binary-tree with at least $N$ leaf nodes, which denote $N$ children users. Set a state $ST := BT$. Output $RL$, $ST$, $PP := \{\mathbf{A}, \bar{\mathbf{A}}, \mathbf{B}_1, \cdots, \mathbf{B}_l, \mathbf{D}_1, \cdots, \mathbf{D}_l, \mathbf{u}\}$, $msk := \{\mathbf{T_A}, \mathbf{T}_{\bar{\mathbf{A}}}\}$.

**PriKeyGen**$(PP, msk, ST, id)$ : On input the public parameter $PP$, the master key $msk$, the state $ST$ and an identity $id = (b_1, \cdots, b_l) \in \{-1, 1\}^l$. It picks an unassigned leaf node $v_{id}$ from $ST$ and stores $id$ in that node. For any $\theta \in \text{Path}(BT, v_{id})$, if $\mathbf{u}_{\theta,id}$, $\mathbf{u}_{\theta,t}$ are undefined, then $\mathbf{u}_{\theta,id} \leftarrow \mathbb{Z}_q^n$, $\mathbf{u}_{\theta,t} := \mathbf{u} - \mathbf{u}_{\theta,id}$. Store $\mathbf{u}_{\theta,id}$, $\mathbf{u}_{\theta,t}$ in node $\theta$ and update a state $ST$. Sample $\mathbf{e}_{\theta,id} \leftarrow \mathbf{SampleLeft}(\mathbf{A}, \Sigma_{i=1}^l b_i\mathbf{B}_i + \mathbf{G}, \mathbf{T_A}, \mathbf{u}_{\theta,id}, \sigma)$ for $\theta \in \text{Path}(BT, v_{id})$. Run $\mathbf{T}_{[\bar{\mathbf{A}}|\Sigma_{i=1}^l b_i\mathbf{B}_i+\mathbf{G}]} \leftarrow \mathbf{ExtRndLeft}(\bar{\mathbf{A}}, \Sigma_{i=1}^l b_i\mathbf{B}_i + \mathbf{G}, \mathbf{T}_{\bar{\mathbf{A}}}, \bar{\sigma})$. Output a private key $sk_{id} := (\{(\theta, \mathbf{e}_{\theta,id})\}_{\theta\in\text{path}(v_{id})}, \mathbf{T}_{[\bar{\mathbf{A}}|\Sigma_{i=1}^l b_i\mathbf{B}_i+\mathbf{G}]})$ and a state $ST$.

**KeyUpd**$(PP, msk, ST, RL, t)$ : On input the master key $msk$, the state $ST$, the revocation list $RL$ and a time $t = (t_1, \cdots, t_l) \in \{-1, 1\}^l$. For any $\theta \in \text{KUNodes}(BT, RL, t)$, if $\mathbf{u}_{\theta,id}$, $\mathbf{u}_{\theta,t}$ are undefined, then $\mathbf{u}_{\theta,t} \leftarrow \mathbb{Z}_q^n$, $\mathbf{u}_{\theta,id} := \mathbf{u} - \mathbf{u}_{\theta,t}$. Store $\mathbf{u}_{\theta,id}$, $\mathbf{u}_{\theta,t}$ in node $\theta$. Sample $\mathbf{e}_{\theta,t} \leftarrow \mathbf{SampleLeft}$ for $\theta \in \text{KUNodes}(BT, RL, t)$. Output a key update $ku_t := \{(\theta, \mathbf{e}_{\theta,t})\}_{\theta\in\text{KUNodes}(BT,RL,t)}$.

**DecKeyGen**$(sk_{id}, ku_t)$ : On input a private key $sk_{id}$ and a key update $ku_t$. Extract $P = \text{Path}(BT, v_{id})$ in $sk_{id}$, and $K = \text{KUNodes}(BT, RL, t)$ in $ku_t$. If $P \cap K = \emptyset$, output $\perp$. Otherwise for the unique node $\theta^* \in P \cap K$, set $\mathbf{dk}_{id,t} = [\mathbf{e}^L_{\theta^*,id} + \mathbf{e}^L_{\theta^*,t} | \mathbf{e}^R_{\theta^*,id} | \mathbf{e}^R_{\theta^*,t}]$. Run $\bar{\mathbf{dk}}_{id,t} \leftarrow \mathbf{SampleLeft}([\bar{\mathbf{A}}|\Sigma_{j=1}^l b_j\mathbf{B}_j + \mathbf{G}|\Sigma_{j=1}^l t_j\mathbf{D}_j + \mathbf{G}], \mathbf{u}, \mathbf{T}_{[\bar{\mathbf{A}}|\Sigma_{j=1}^l b_j\mathbf{B}_j+\mathbf{G}]}, \sigma)$. Output a decryption key $dk_{id,t} := (\mathbf{dk}_{id,t}, \bar{\mathbf{dk}}_{id,t}) \in \mathbb{Z}^{3m} \times \mathbb{Z}^{3m}$.

**Encrypt**$(PP, id, t, m)$ : On input the parameter $PP$, a time $t = (t_1, \cdots, t_l) \in \{-1, 1\}^l$, an identity $id = (b_1, \cdots, b_l) \in \{-1, 1\}^l$ and a message $m$. Construct $\mathbf{A}_{id,t} := [\mathbf{A}|\Sigma_{i=1}^l b_i\mathbf{B}_i + \mathbf{G}|\Sigma_{i=1}^l t_i\mathbf{D}_i + \mathbf{G}]$, and $\bar{\mathbf{A}}_{id,t} := [\bar{\mathbf{A}}|\Sigma_{i=1}^l b_i\mathbf{B}_i + \mathbf{G}|\Sigma_{i=1}^l t_i\mathbf{D}_i + \mathbf{G}]$. Select uniformly random vectors $\mathbf{s}, \bar{\mathbf{s}} \xleftarrow{s} \mathbb{Z}_q^n$. Sample error vectors $e_0 \leftarrow D_{\mathbb{Z},\alpha q}$, $\mathbf{e}_1, \bar{\mathbf{e}}_1 \leftarrow D_{\mathbb{Z}^{3m},\alpha' q}$ and set $c_0 = \mathbf{u}^T(\mathbf{s} + \bar{\mathbf{s}}) + e_0 + m\lfloor\frac{q}{2}\rfloor$, $\mathbf{c}_1 = \mathbf{A}^T_{id,t}\mathbf{s} + \mathbf{e}_1$, $\bar{\mathbf{c}}_1 = \bar{\mathbf{A}}^T_{id,t}\bar{\mathbf{s}} + \bar{\mathbf{e}}_1$. Output a ciphertext $ct_{id,t} := (c_0, \mathbf{c}_1, \bar{\mathbf{c}}_1) \in \mathbb{Z}_q \times \mathbb{Z}_q^{3m} \times \mathbb{Z}_q^{3m}$.

**ReKeyGen**, **ReEncrypt**, **ReEncVer**, **Decrypt** : Similar to the above construction.

The proof process is in the full version.

## References


1. Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 127–144. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054122
2. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_28

3. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans. Inf. Syst. Secur. **9**(1), 1–30 (2006)
4. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. SIAM J. Comput. **32**(3), 586–615 (2003)
5. Boldyreva, A., Goyal, V., Kumar, V.: Identity-based encryption with efficient revocation. In: Proceedings of the 15th ACM Conference on Computer and Communications Security, pp. 417–426 (2008)
6. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. J. Cryptol. **25**(4), 601–639 (2012)
7. Dutta, P., Susilo, W., Duong, D.H., Baek, J., Roy, P.S.: Lattice-based unidirectional IBPRE secure in standard model. arXiv preprint arXiv:2005.06741 (2020)
8. Dutta, P., Susilo, W., Duong, D.H., Roy, P.S.: Collusion-resistant identity-based proxy re-encryption: lattice-based constructions in standard model. Theoret. Comput. Sci. **871**, 16–29 (2021)
9. Green, M., Ateniese, G.: Identity-based proxy re-encryption. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 288–306. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72738-5_19
10. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, pp. 197–206 (2008)
11. Gorbunov, S., Vaikuntanathan, V., Wichs, D.: Leveled fully homomorphic signatures from standard lattices. In: Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, pp. 469–477 (2015)
12. Katsumata, S., Matsuda, T., Takayasu, A.: Lattice-based revocable (hierarchical) IBE with decryption key exposure resistance. Theoret. Comput. Sci. **809**, 103–136 (2020)
13. Katsumata, S., Yamada, S.: Partitioning via non-linear polynomial functions: more compact IBEs from ideal lattices and bilinear maps. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 682–712. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_23
14. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41
15. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 1–40 (2008)
16. Singh, K., Rangan, C.P., Banerjee, A.K.: Lattice based identity based proxy re-encryption scheme. J. Internet Serv. Inf. Secur. **3**(3/4), 38–51 (2013)
17. Singh, K., Rangan, C.P., Banerjee, A.K.: Lattice based identity based unidirectional proxy re-encryption scheme. In: Chakraborty, R.S., Matyas, V., Schaumont, P. (eds.) SPACE 2014. LNCS, vol. 8804, pp. 76–91. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12060-7_6

# Malicious Domain Detection with Heterogeneous Graph Propagation Network

Cheng Hu[1,2], Fangfang Yuan[1(✉)], Yanbing Liu[1,2], Cong Cao[1], Chunyan Zhang[1], and Jianlong Tan[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{hucheng,yuanfangfang,liuyanbing,caocong,zhangchunyan,tanjianlong}@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

**Abstract.** As one of the most important basic services of the Internet, the domain name system is abused by attackers for various malicious activities. Malicious domain detection is a key technology against attackers. Previous works mainly employ manually selected features to detect malicious domains which are easily evaded by attackers. In this paper, we propose a novel malicious domain detection system with heterogeneous graph propagation network, named HGPNDom, which can jointly consider the global relationship and higher-order features of domains. In HGPNDom, we first model the DNS scene as a heterogeneous information network (HIN) to capture rich information. Then, we propose a heterogeneous graph propagation network (HGPN) to classify domain nodes in the HIN, including semantic propagation mechanism and semantic fusion mechanism. The semantic propagation mechanism can spread information through more layers and learn higher-order domain features, while the semantic fusion mechanism can learn the importance of different meta-paths and fuse them for classification. Experimental results on the real DNS dataset show that HGPNDom outperforms other state-of-the-art methods.

**Keywords:** Malicious domain detection · Heterogeneous graph neural network · Heterogeneous information network

## 1 Introduction

Domain Name System (DNS) contains distributed servers, provides services that map easy-to-remember domains to IP addresses, allows users to easily locate devices, services, or other resources on the Internet. In recent years, due to its cheapness and flexibility, domains have been used for various malicious activities. For example, malicious domains are used to facilitate command and control communications or host phishing webpages, often causing economic losses and


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 545–556, 2022.
https://doi.org/10.1007/978-3-031-19208-1_45

privacy data leakage. How to effectively fight against attackers and detect malicious domains have always been a hot topic in the field of cyberspace security.

The most traditional method of malicious domain detection is to use blacklists and rules [5,10]. However, with the rise of Domain-Flux and Fast-Flux techniques, the rule-based methods become infeasible. To solve this issue, some researchers propose the feature-based methods to identify malicious domains [1–3,11]. They extract features from DNS traffic, construct a machine learning-based classifier to distinguish malicious and benign domains. But these methods treat each domain separately, do not consider the relationships among domains and attackers can evade them by tampering with domain features. Recently, researchers find that it is difficult for attackers to forge the relationship between domains. Hence, they utilize associations between domains to construct a graph [13,14] and use graph neural network (GNN) to recognize malicious domains. Nevertheless, these GNN-based methods have the following drawbacks: (1) Only use a fraction of layers to propagate information and only aggregate the information of limited neighbor nodes, so they cannot capture the complex implicit relationships between domains. (2) With the increase of the number of layers, the number of aggregated neighbor nodes increases drastically. Eventually, the information aggregated by each node is the same, which suffers from the oversmoothing problem.

In order to address the above problems, this paper proposes a malicious domain detection system with heterogeneous graph propagation network, named HGPNDom. Specifically, We firstly model the DNS scene as a heterogeneous information network with three types of nodes: hosts, domains, and IP addresses. Secondly, we propose a heterogeneous graph propagation network, named HGPN, which includes semantic propagation mechanism and semantic fusion mechanism. The semantic propagation mechanism can consider the characteristics of the domain node itself when aggregating meta-path based neighbors through the attention mechanism, which alleviates the oversmoothing problem. Then, the semantic fusion mechanism can learn the importance of different meta-paths and fuse them to get the final domain representation for malicious domain detection. Finally, we validate the effectiveness of HGPNDom on the real-world DNS dataset. The main contributions of our paper are as follows:

(1) We analyze the oversmoothing problem in the GNN-based detection methods and first propose a novel Heterogeneous Graph Propagation Network (HGPN) for malicious domain detection, which can alleviate oversmoothing problem.
(2) In order to learn more representative domain node embedding, we use the attention mechanism in both the semantic propagation mechanism and the semantic fusion mechanism, which can learn the importance of different neighbor nodes and the importance of different meta-paths to the current domain node.
(3) We implement a prototype of HGPNDom and test it on real DNS dataset, experimental results demonstrate the effectiveness of our proposed method.

The rest of this paper is divided as follows. Section 2 introduces the necessary concepts. We describe the malicious domain detection method proposed in Sect. 3. In Sect. 4, we conduct experiments to verify the effectiveness of the proposed method. In Sect. 5, we introduce the related work and Sect. 6 summarizes the work of this paper.

## 2 Preliminaries

### 2.1 Heterogeneous Information Network

A directed graph is defined as $G = (v, \varepsilon)$ [16], $v$ represents the set of nodes in the graph, $\varepsilon$ represents the set of edges in the graph. The graph also has a node mapping function $\phi : v \rightarrow A$ and an edge mapping function $\varphi : \varepsilon \rightarrow R$. $A$ and $R$ are a collection of node types and edge types, respectively. If $|A| + |R| > 2$, it means that there are different types of nodes or edges in the graph, such graphs are called heterogeneous graphs, namely heterogeneous information networks.

### 2.2 Meta-path

Two objects in a HIN can be connected by different kinds of paths. A meta-path $\Phi$ is defined as a path of the form $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{l+1}$ [16], which describes a composite relation $R = R_1 \circ R_2 \circ \ldots \circ R_l$ between objects $A_1, A_2, \ldots, A_{l+1}$, where $\circ$ represents a composition operator on the relation.

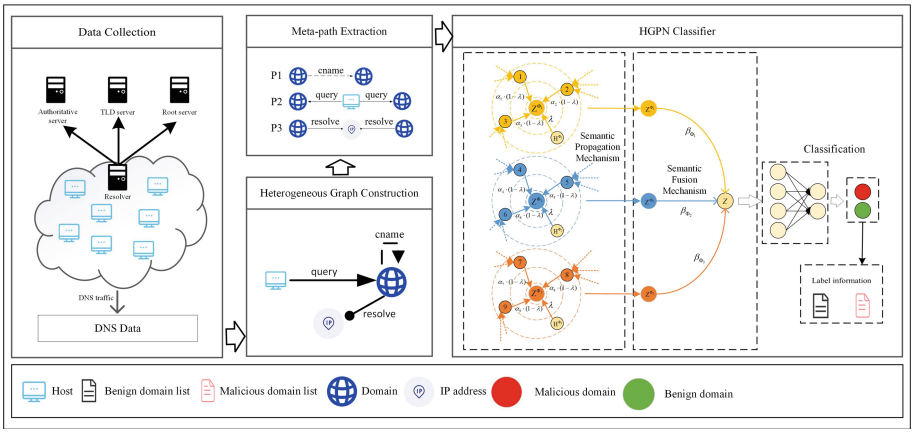



**Fig. 1.** Overall Framework of HGPNDom.

## 3 The Proposed Method

Figure 1 shows the overall framework of HGPNDom, which includes four parts: data collection, heterogeneous graph construction, meta-path extraction and HGPN classifier. This section will describe each part in detail.

### 3.1 Data Collection

In order to reflect the real DNS scene and obtain more representative DNS data, the DNS dataset we collected includes three types of nodes: hosts, domains, and IP addresses and three types of relationships: the request relationship between hosts and domains, the resolution relationship between domains and IP addresses, and the CNAME record between domains and domains.

### 3.2 Heterogeneous Graph Construction

We model the DNS scene as a heterogeneous graph based on the collected DNS data. Since the real-word DNS data contains a lot of noise, in order to improve the training time of the model, the noise nodes need to be pruned. We prune the constructed heterogeneous graph based on the following six strategies:

- Inactive hosts: The number of domains queried by some hosts is less than $K_f$, these hosts are inactive and have little contribution to identify malicious domains.
- Large hosts: Some hosts query more than $K_c\%$ of the total domains, these hosts may be proxies.
- Popular domains: Some domains have been accessed by more than $K_q\%$ of the total number of hosts, if these domains are malicious, they will have a serious impact and will be easily discovered by the network security management department.
- Irregular domains: Some domains are visited less than $K_l$ times, and these domains do not provide enough information.
- Rare IPs: Some IP addresses are only mapped to a domain, and these IP addresses do not contribute much to label delivery.
- Rare domains: Some domains are only resolved to an IP address, which makes it difficult to pass labels.

### 3.3 Meta-path Extraction

Meta-path is a common method for semantic extracting of heterogeneous graphs, each meta-path represents a specific semantic. As shown in Fig. 1, we extract three symmetric meta-paths for malicious domain detection.

The meta-path P1 indicates that two different domains belong to the same CNAME record. According to relevant facts, the cname domain of a malicious domain has a high probability of being a malicious domain, vice versa. The meta-path P2 represents the query relationship between hosts and domains. We believe that two hosts attacked by the same attacker will have overlapping sets of malicious domains, and the probability of a normal host querying the same malicious domain is low. The meta-path P3 indicates the mapping relationship between domains and IP addresses. As the number of IP addresses is relatively stable in the Internet, domains that are resolved to the same IP address tend to have the same category over a period of time.

### 3.4 HGPN Classifier

We design a heterogeneous graph propagation network (HGPN) to classify domain nodes in the HIN. As shown in Fig. 1, HGPN classifier includes feature extract, semantic propagation mechanism, semantic fusion mechanism and classification function. The semantic propagation mechanism can alleviate the oversmoothing problem at the node level, while the semantic fusion mechanism can learn the importance of each meta-path and fuse different meta-paths by weight. Then, the classification function can detect whether the domain node is a benign domain or a malicious domain.

**Feature Extract.** We initialize the domain nodes in the HIN which constructed in Sect. 3.2. We refer to FANCI [11] and extract 21 features as the initial feature vector for each domain node.

**Semantic Propagation Mechanism.** For each meta-path $\Phi$ in the HIN, the semantic propagation mechanism uses the semantic aggregation function $g_\Phi$ to aggregate the neighbor node features based on the meta-path, and learn a semantic-specific node embedding, shown as follows:

$$\mathbf{Z}^\Phi = g_\Phi(\mathbf{X}), \tag{1}$$

where $\mathbf{X}$ denotes the initial feature matrix, $\mathbf{Z}^\Phi$ denotes the specific semantic embedding learned under the meta-path $\Phi$. In order to alleviate the oversmoothing problem at the node level, a semantic aggregation function is designed, shown as follows:

$$e_{ij}^\Phi = att_{node}(\mathbf{h}_i, \mathbf{h}_j; \Phi), \tag{2}$$

$$\alpha_{ij}^\Phi = \frac{\exp(e_{ij}^\Phi)}{\sum_{j \in N_i^\Phi} \exp(e_{ij}^\Phi)}, \tag{3}$$

$$\mathbf{Z}^{\Phi,k} = g_\Phi(\mathbf{Z}^{\Phi,k-1}) = (1-\lambda) \cdot \boldsymbol{\alpha}^\Phi \cdot \mathbf{Z}^{\Phi,k-1} + \lambda \cdot \mathbf{H}^\Phi, \tag{4}$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ denote the feature of node $i$ and $j$, respectively. $e_{ij}^\Phi$ denotes the importance of node $j$ to node $i$, node pair $(i, j)$ are connected by meta-path $\Phi$, $att_{node}$ denotes the deep neural network that learns node-level attention. And $\alpha_{ij}^\Phi$ denotes the normalization of $e_{ij}^\Phi$ via softmax function, $N_i^\Phi$ denotes the meta-path $\Phi$ based neighbors of node $i$, $\boldsymbol{\alpha}^\Phi$ denotes the attention vector composed of $\alpha_{ij}^\Phi$. Note that $\boldsymbol{\alpha}^\Phi \cdot \mathbf{Z}^\Phi$ denotes aggregating meta-path $\Phi$ based neighbor nodes and $\mathbf{H}^\Phi$ denotes the characteristics of each node itself. Here $\lambda$ is the weight scalar, which denotes the importance of each node's characteristics in the aggregation process. $\mathbf{Z}^{\Phi,k}$ is the node embedding learned through the k-layer semantic propagation mechanism and we treat it as a learned semantic-specific node embedding based on the meta-path $\Phi$.

**Semantic Fusion Mechanism.** We design three meta-paths which are described in Sect. 3.3. Each meta-path represents a specific semantic information and the node embedding of a specific semantic can only reflect the node information from one perspective. In order to describe the node characteristics more comprehensively, the semantic fusion mechanism integrates multiple meta-paths to capture the rich semantic information on heterogeneous graphs and reflect it on node embedding. We use semantic fusion mechanism $F$ to aggregate the node embeddings of P-group specific semantics for malicious domain detection, shown as follows:

$$\mathbf{Z} = F(\mathbf{Z}^{\Phi_1}, \mathbf{Z}^{\Phi_2}, \ldots, \mathbf{Z}^{\Phi_P}), \tag{5}$$

where $\{\Phi_1, \Phi_2, \ldots, \Phi_P\}$ denotes a set of meta-paths, $\{\mathbf{Z}^{\Phi_1}, \mathbf{Z}^{\Phi_2}, \ldots, \mathbf{Z}^{\Phi_P}\}$ denotes the node embeddings of P-group specific semantics. The semantic fusion mechanism first projects different semantics into the same space and adopts semantic fusion vector $\mathbf{q}$ to learn the importance of different meta-paths. Then, it normalizes the weight of each meta-path, and fuses each semantics to get the final node embedding $\mathbf{Z}$, shown as follows:

$$w_{\Phi_p} = \frac{1}{|v|} \sum_{i \in v} \mathbf{q}^T \cdot \tanh(\mathbf{W} \cdot \mathbf{z}_i^{\Phi_p} + \mathbf{b}), \tag{6}$$

$$\beta_{\Phi_p} = \frac{\exp(w_{\Phi_p})}{\sum_{p=1}^{P} \exp(w_{\Phi_p})}, \tag{7}$$

$$\mathbf{Z} = \sum_{p=1}^{P} \beta_{\Phi_p} \cdot \mathbf{Z}^{\Phi_p}, \tag{8}$$

where $\mathbf{W}$ denotes the weight matrix, $\mathbf{b}$ denotes the bias vector, $w_{\Phi_p}$ denotes the weight of meta-path $p$, $\beta_{\Phi_p}$ denotes the normalization of $w_{\Phi_p}$.

**Classification.** With the final node embedding $\mathbf{Z}$, the domain classification task can be transformed into a binary classification task. We use a fully connected network to classify domains, shown as follows:

$$\hat{\mathbf{y}} = \sigma(\mathbf{W} \cdot \mathbf{Z} + \mathbf{b}), \tag{9}$$

where $\hat{\mathbf{y}}$ denotes the prediction probability. $\mathbf{W}$, $\mathbf{b}$ denote the weight matrix and bias vector, respectively. $\sigma$ denotes the ReLU activation function. We calculate Cross Entropy and update parameters in HGPN classifier, shown as follows:

$$L = -\sum_{l \in \mathcal{Y}_\mathcal{L}} \mathbf{Y}_l \cdot \ln(\hat{\mathbf{y}}), \tag{10}$$

where $\mathcal{Y}_\mathcal{L}$ denotes the set of labelled nodes, $\mathbf{Y}_l$ is the label vector. Under the guide of the labelled data, we can optimize the proposed model for malicious domain detection.

# 4 Experiments

## 4.1 Dataset

We collect DNS traffic data from a university LAN for two weeks from 2020.8.31 to 2020.9.13, as the real DNS data used in the experiment. Firstly, we parse the DNS data to obtain the information like domains, hosts, IP addresses and their relationships. Secondly, We remove some noisy nodes through the pruning strategy in Sect. 3.2. Finally, we build the DNS HIN that contains 4,330,702 domains, 251,518 hosts, 433,702 IP addresses and edges between them. The statistic information of experimental dataset is shown in Table 1.

Since our approach is based on semi-supervised learning, we need the labelled dataset. For benign domains, we select the domains whose 2LD appears in the Alexa Top1M list. For malicious domains, we merge the domains in Malwaredomains.com, phishtank and other lists as the malicious domain list and use the 2LD of domains to match. In addition, we use the VirusTotal and Google Safe Browsing to validate the domains. In the end, we obtain 20,000 benign domains and 20,000 malicious domains. We randomly divide the labelled dataset into train set, validation set and test set according to the ratio of 3:1:1.

**Table 1.** Statistics of experimental dataset

| DNS HIN | | |
|---|---|---|
| #Domains | #Hosts | #IPs |
| 4,330,702 | 251,518 | 433,702 |
| #Domain-Domain | #Domain-IP | #Domain-Host |
| 30,792 | 1,318,085 | 4,403,977 |

## 4.2 Parameter Setting and Evaluation Metrics

According to the pruning strategy in Sect. 3.2, we set $K_f = 2$, $K_c = 80$, $K_q = 50$, $K_l = 100$. We leverage Adam to update parameters and set the learning rate to 0.01. Relying on experience, we set the final embedding dimension to 32 for all the methods. We use F1 score, Precision and Recall as the evaluation metrics.

## 4.3 Performance Evaluation of HGPN

We select five graph representation learning methods for malicious domain detection to verify the effectiveness of HGPN, including two homogeneous graph neural network methods, two heterogeneous graph embedding methods and one heterogeneous graph neural network method. Methods are list as follows:

**GAT** [15]: A homogeneous graph neural network with attention mechanism. Here we test all the meta-paths for GAT and report the best performance.

**PPNP** [8]: A homogeneous graph neural network with personalized propagation scheme. Here we test all the meta-paths for PPNP and report the best performance.

**Metapath2vec** [4]: A heterogeneous graph embedding method which uses meta-path based random walk and skip-gram to learn node embeddings. Here we test all the meta-paths for metapath2vec and report the best performance.

**HERec** [12]: A heterogeneous graph embedding method which designs a constraint strategy to filter node sequences and uses skip-gram for graph embedding. Here we test all the meta-paths for HERec and report the best performance.

**HGT** [6]: A heterogeneous graph neural network based on heterogeneous mutual attention which aggregates information through meta-relational triples.

Table 2 shows the comparison results of the graph representation learning methods. We see that the graph neural network methods outperform the graph embedding methods Metapath2vec and HERec. It is because that graph neural network methods consider the global relationship and node features at the same time. For homogeneous graph neural network methods, GAT can learn the weight of each node, and its performance is better than PPNP. For heterogeneous graph neural network methods, the performance of HGPN outperforms HGT, because it can propagate through more layers, aggregate more neighbor nodes. Overall, HGPN outperforms all other graph representation methods, as it can jointly consider the global relationship and alleviate oversmoothing problem, get more representative embeddings for classification.

**Table 2.** Performance of HGPN

| Method | F1 Score (%) | Precision (%) | Recall (%) |
|---|---|---|---|
| Metapath2vec | 52.93 | 55.57 | 50.52 |
| HERec | 52.35 | 56.49 | 48.47 |
| GAT | 92.33 | 89.70 | **95.13** |
| PPNP | 85.58 | 87.25 | 83.97 |
| HGT | 88.99 | 88.08 | 89,92 |
| HGPN | **95.38** | **95.77** | 95.00 |

Furthermore, we study the impact of different meta-paths on malicious domain detection. Table 3 shows the weights assigned to each meta-path in HGPN's semantic fusion mechanism. P1 has the highest attention weight, which is consistent with the fact that domains with the same CNAME record tend to have the same category. The attention weights of P2 and P3 are similar, and both are relatively low. It may be due to the existence of the DHCP protocol in the network that the IP address of the host is not fixed, and the IP address of domain resolution changes dynamically due to technologies such as Domain-Flux and Fast-Flux.

**Table 3.** Attention weights of different meta-paths

| ID | Meta-path | Attention weight |
|---|---|---|
| P1 | DD | 0.7704 |
| P3 | DPD | 0.1159 |
| P2 | DHD | 0.1137 |

### 4.4 Ablation Studies

In order to better understand the contribution of each part of HGPN, we separately remove the semantic propagation mechanism and semantic fusion mechanism of HGPN, and observe the changes in model performance.

When the semantic propagation mechanism is removed, the $\lambda$ is set to 0. Table 4 shows that when the semantic propagation mechanism is removed, F1 Score of HGPN decreases by 9.24%, indicating that the semantic propagation mechanism can capture higher-order domain features and learn better domain representation.

To remove the semantic fusion mechanism is to simply average all meta-paths when aggregating different semantics. Table 4 shows that when the semantic fusion mechanism is removed, F1 Score of HGPN decreases by 2.47%. It shows that different meta-paths denote different semantics and the semantic fusion mechanism can give appropriate weights to them.

**Table 4.** Effect of HGPN

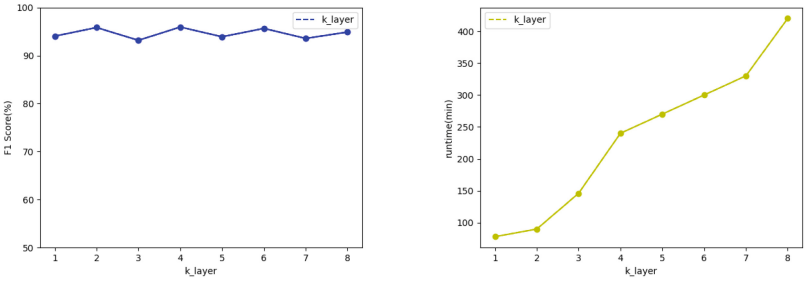
| | F1 Score (%) | Precision (%) | Recall (%) |
|---|---|---|---|
| (−)Semantic Propagation Mechanism | 86.14(−9.24) | 83.57(−12.2) | 88.88(−6.12) |
| (−)Semantic Fusion Mechanism | 92.91(−2.47) | 93.27(−2.5) | 92.55(−2.45) |

### 4.5 Parameter Sensitivity

We investigate the sensitivity of HGPN to two parameters: the number of the layers k_layer and the weight scalar $\lambda$.

Figure 2(a) shows the influence of k_layer on F1 Score. With the increase of k_layer, F1 Score fluctuates between 92% and 96%, indicating that the semantic propagation mechanism can effectively alleviate oversmoothing problem. When the value of k_layer is 2, 4 and 6, respectively, HGPN gets better results. Figure 2(b) shows the relationship between k_layer and runtime. With the increase of k_layer, the runtime also increases. When k_layer is greater than 2, the runtime increases rapidly. Considering the model effect and running efficiency, k_layer is finally set to 2.

Figure 3 shows the relationship between the weight scalar $\lambda$ and F1 Score. With the increase of $\lambda$, the effect of HGPN shows a downward trend, but it remains above 93%. When $\lambda$ is 0.1, the performance of HGPN is the best. Therefore, the weight scalar $\lambda$ is finally set to 0.1.




(a) Impact of F1 Score (b) Impact of runtime

**Fig. 2.** k_layer analysis

### 4.6 Performance Evaluation of HGPNDom

In order to verify the effectiveness of HGPNDom, we compare it with the feature-based malicious domain detection system FANCI and the association-based malicious domain detection system DeepDom. The two systems are described as follows:

**FANCI** [11]: FANCI is a feature-based detection system. It extracts a total of 21 features from three aspects: structure features, linguistic features, and statistical features. Finally, Support Vector Machine or Random Forest are used to classify domains.

**DeepDom** [13]: DeepDom is the latest association-based detection system, which models the DNS scene as a heterogeneous graph, uses SHetGCN to realize domain classification. Sun et al. [13] analyse the impact of different meta-paths on malicious domain detection, and find that the contribution of three meta-paths: domain-domain, domain-host-domain, and domain-IP-domain accounted for nearly 80%. When we construct the heterogeneous graph, we only consider the three meta-paths with the highest weights.

Table 5 shows the detection results of each system. We can see that our proposed HGPNDom outperforms FANCI and DeepDom. The reason is that FANCI treats each domain separately without considering the relationship between domains, which will cause some information loss and lead to poor generalization ability. SHetGCN used by DeepDom fails to solve the oversmoothing problem of the graph neural network, and the detection effect is slightly lower than that of HGPNDom.

**Table 5.** Comparison with other malicious domain detection systems

| Method | F1 Score (%) | Precision (%) | Recall (%) |
|---|---|---|---|
| FANCI | 78.37 | 81.21 | 75.73 |
| DeepDom | 94.32 | 94.78 | 93.87 |
| HGPNDom | **95.38** | **95.77** | **95.00** |

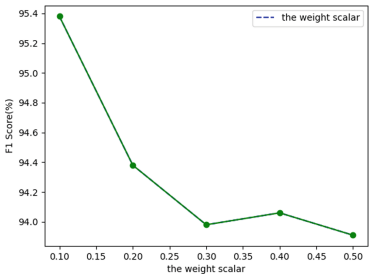



**Fig. 3.** The weight scalar $\lambda$ analysis

## 5 Related Work

Malicious domain detection methods are mainly divided into three categories: rule-based methods, feature-based methods and association-based methods. The rule-based methods are the most traditional methods and rely heavily on experience [5]. It's hard to fight against ever-evolving cyberattacks. The feature-based methods build classifiers based on features which are extracted from DNS traffic and use classifiers to detect malicious domains [1,3,11]. The feature-based methods do not consider the relationship between domains and rely on manual features, this methods are easily evaded by sophisticated attackers. The association-based methods considers the relationship between domains, which is difficult for attackers to falsify. Researchers model the DNS scene as a graph [7,9,13] and use graph neural network (GNN) to recognize malicious domains. Due to the oversmoothing problem of GNN, each domain node cannot adequately aggregate neighbor nodes.

## 6 Conclusion

This paper proposes a novel malicious domain detection system HGPNDom. Considering the oversmoothing problem of the GNN-based malicious domain detection methods, we propose HGPN, which includes semantic propagation mechanism and semantic fusion mechanism. Absorbing the characteristics of the node itself when aggregating meta-path based neighbors, HGPN can learn more representative node embeddings with more propagation layers. Experimental results show that HGPNDom outperforms other state-of-the-art malicious domain detection systems.

**Acknowledgements.** This work was partly supported by Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No. XDC02030000.

## References

1. Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., Feamster, N.: Building a dynamic reputation system for {DNS}. In: 19th USENIX Security Symposium (USENIX Security 10) (2010)
2. Antonakakis, M., Perdisci, R., Lee, W., Vasiloglou II, N., Dagon, D.: Detecting malware domains at the upper {DNS} hierarchy. In: 20th USENIX Security Symposium (USENIX Security 11) (2011)
3. Bilge, L., Kirda, E., Kruegel, C., Balduzzi, M.: Exposure: finding malicious domains using passive DNS analysis. In: Ndss, pp. 1–17 (2011)
4. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 135–144 (2017)
5. Grill, M., Nikolaev, I., Valeros, V., Rehak, M.: Detecting DGA malware using NetFlow. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 1304–1309. IEEE (2015)
6. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: Proceedings of the Web Conference 2020, pp. 2704–2710 (2020)
7. Khalil, I., Yu, T., Guan, B.: Discovering malicious domains through passive DNS data graph analysis. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, pp. 663–674 (2016)
8. Klicpera, J., Bojchevski, A., Günnemann, S.: Predict then propagate: graph neural networks meet personalized pagerank. arXiv preprint arXiv:1810.05997 (2018)
9. Manadhata, P.K., Yadav, S., Rao, P., Horne, W.: Detecting malicious domains via graph inference. In: Kutyłowski, M., Vaidya, J. (eds.) ESORICS 2014. LNCS, vol. 8712, pp. 1–18. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11203-9_1
10. Sato, K., Ishibashi, K., Toyono, T., Hasegawa, H., Yoshino, H.: Extending black domain name list by using co-occurrence relation between DNS queries. IEICE Trans. Commun. **95**(3), 794–802 (2012)
11. Schüppen, S., Teubert, D., Herrmann, P., Meyer, U.: {FANCI}: feature-based automated {NXDomain} classification and intelligence. In: 27th USENIX Security Symposium (USENIX Security 18), pp. 1165–1181 (2018)
12. Shi, C., Hu, B., Zhao, W.X., Philip, S.Y.: Heterogeneous information network embedding for recommendation. IEEE Trans. Knowl. Data Eng. **31**(2), 357–370 (2018)
13. Sun, X., Wang, Z., Yang, J., Liu, X.: Deepdom: malicious domain detection with scalable and heterogeneous graph convolutional networks. Comput. Secur. **99**, 102057 (2020)
14. Sun, X., Yang, J., Wang, Z., Liu, H.: HGDom: heterogeneous graph convolutional networks for malicious domain detection. In: NOMS 2020 IEEE/IFIP Network Operations and Management Symposium, pp. 1–9. IEEE (2020)
15. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
16. Wang, X., et al.: Heterogeneous graph attention network. In: The World Wide Web Conference, pp. 2022–2032 (2019)

# Interference Mitigation via Collaborative Beamforming in UAV-Enabled Data Collections: A Multi-objective Optimization Method

Hongjuan Li, Da Wei, Geng Sun(✉), Jian Wang, Jiahui Li, and Hui Kang

College of Software and College of Computer Science and Technology, Jilin University, Changchun 130012, China
sungeng@jlu.edu.cn

**Abstract.** Unmanned aerial vehicles (UAVs) are adopted as promising platforms to provide aerial wireless communications and networks. However, due to the line-of-sight (LoS) dominant air-ground channels, UAVs cause stronger interference to the terrestrial network devices. In this work, we study a novel interference mitigation method via collaborative beamforming (CB) under a UAV-enabled data collection scenario. Specifically, we form a UAV-enabled virtual antenna array (UVAA) to transmit the collected data to the terrestrial base stations (BSs), and formulate an interference mitigation multi-objective optimization problem (IMMOP) to simultaneously enhance the data transmission efficiency, reduce the interference affection and increase the network lifetime. Due to the complexity and NP-hardness of IMMOP, a chaotic multi-objective multi-verse optimizer (CMOMVO) is proposed for solving the problem. Simulation results show that the CMOMVO can effectively solve the IMMOP and has better performance than some benchmark algorithms.

**Keywords:** Collaborative beamforming · Interference mitigation · Multi-objective optimization · UAV communications

## 1 Introduction

Due to their inherent merits, unmanned aerial vehicles (UAVs) have been widely employed in various military, civilian and commercial applications [17]. Recently, UAV communications and networks are regarded as vital components in the promising 5G/6G networks [15,19]. Rely on the line-of-sight (LoS) dominant air-ground channels, UAVs can be deployed as aerial base stations (BSs) to provide efficient and low-cost network services for the terrestrial terminals [16]. Moreover, UAVs can be dispatched to a remote place for automated data collections after designing their locations and trajectories [4,5].

Despite the considerable benefits of LoS dominant channels, UAV-enabled data collection also faces some key issues. In particular, UAV networks may


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 557–566, 2022.
https://doi.org/10.1007/978-3-031-19208-1_46

generate stronger interference to terrestrial terminals in the LoS channel, which can degrade the performance of the terrestrial networks. Thus, various interference mitigation methods such as power control and three-dimensional (3D) trajectory design in the literature have been proposed for solving the interference problems [6,7]. However, the trajectory and power allocation may decrease the wireless communication ability of the UAV-enabled data collection systems since the transmission distance and power may be increased and decreased, respectively. Thus, it is necessary to study a novel interference mitigation approach to reduce interference while improving the data collection efficiency.

Collaborative beamforming (CB) is an effective method that can significantly enhance wireless communication ability in multi-UAV systems. Moreover, CB can be adopted in UAV networks for high-performance and beneficial communications. For example, the existing works used CB to achieve physical layer security [2], time minimization [12] and energy-efficient communications [14] in UAV networks. However, the randomly distributed UAVs may damage the beam pattern, thus decreasing the performance of the UVAA. Thus, the positions and excitation current weights of the UAVs need to be carefully designed. Moreover, the fine-tuning of the UAVs' positions will lead to extra energy consumption of the UAVs. Thus, we also need to restrict the total propulsion energy consumption of the UAVs.

The main contributions of this work are summarized as follows:

*i)* We consider a typical UAV-enabled data collection scenario of the UAV networks, i.e., the UAVs perform a UVAA to transmit the collected data to the selected BS by using CB. Then, we formulate an interference mitigation multi-objective optimization problem (IMMOP) to simultaneously minimize the data transmission time, mitigate the interference affection and reduce the energy consumption of the UAVs.
*ii)* We propose a chaotic multi-objective multi-verse optimizer (CMOMVO) to solve the formulated IMMOP. Specifically, we enhance the solution initialization, algorithm parameter update and solution update phases via the chaos theory, thus making the algorithm more suitable for solving the IMMOP.
*iii)* Simulation results show that the proposed CMOMVO outmatches other benchmark algorithms for solving the formulated IMMOP.
The rest of this work is arranged as follows. Section 2 presents the models and formulates the IMMOP. Section 3 proposes the CMOMVO. The simulation results are provided in Sect. 4, and the paper is concluded in Sect. 5.

## 2 Models and Problem Formulation

In this section, we first present the models used in this paper and then formulate the IMMOP.

### 2.1 Models

As shown in Fig. 1, a UAV-enabled data collection scenario is considered, in which a set of UAVs denoted as $\mathcal{U} = \{1, 2, \ldots, N_{UAV}\}$ are dispatched to collect

data, while a set of terrestrial BSs denoted as $\mathcal{B} = \{1, 2, \ldots, N_{BS}\}$ can communicate with the UAVs and other ground users (GUs). At a certain time, i.e., the collected data reach the cached threshold, the UAVs will form a UVAA for transmitting the collected data to one of the BS of $\mathcal{B}$. During the process, there are some GUs are communicating with the BSs of $\mathcal{B}$, which means that these communications may be interfered by the UVAA. Without loss of generality, we consider a 3D Cartesian coordinate system, and the positions of the $m$th UAV and the $n$th BS are represented as $(x_m^U, y_m^U, z_m^U)$ and $(x_n^B, y_n^B, 0)$, respectively. Moreover, some key models are detailed as follows.

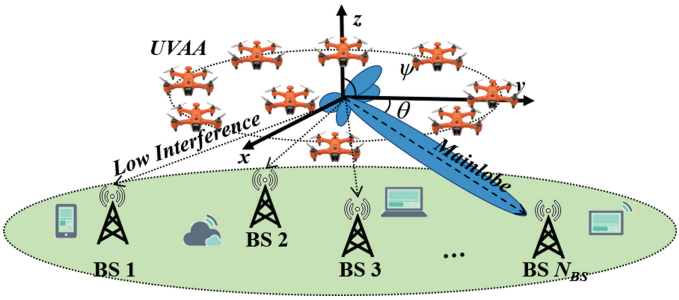



**Fig. 1.** Sketch map of the UAV-enabled data collection system with the CB-based interference mitigation.

**Array Factor of UVAA.** The array factor (AF) can represent the signal intensity of an antenna system in all directions. In the UVAA, the positions and excitation current weights of the UAVs can determine the radiation distribution of AF. Let $\omega_m$ denotes the excitation current weight of the $m$th UAV of the UVAA, the AF can be modeled as follows [1]:

$$AF(\theta, \phi) = \sum_{m=1}^{N_{UAV}} \omega_m e^{j\left[\kappa\left(x_m^U \sin\theta\cos\phi + y_m^U \sin\theta\sin\phi + z_m^U \cos\theta\right)\right]}, \tag{1}$$

where $\theta \in [0, \pi]$ and $\phi \in [-\pi, \pi]$ are the elevation and azimuth angles, respectively, $\kappa = 2\pi/\lambda$ represents the phase constant and $\lambda$ is the wavelength.

**Transmission and Interference Model.** Due to the high altitude of the UAVs, we adopt a LoS channel which may suffer serious interferences. Thus, the transmission rate from the UVAA to the receiver BS is given by [12]:

$$R_{[BS_{rec}]} = B \log_2 \left(1 + \frac{P^{rec}_{[BS_{rec}]}}{\sigma^2}\right), \tag{2}$$

where $P^{rec}_{[BS_{rec}]} = P_t K_0 d^{-\alpha}_{[BS_{rec}]} G_{[BS_{rec}]}$ is the receiver power of the receiver BS from the UVAA, in which $P_t$, $K_0$ and $\alpha$ are the total transmission power of the UVAA, constant pathloss coefficient and pathloss exponent, respectively.

Moreover, $d_{[BS_{rec}]}$ is the Euclidean distance between the UVAA and the receiver BS, and $G_{[BS_{rec}]}$ is the antenna gain of the direction towards the receiver BS, which can be calculated as follows:

$$G_{[BS_{rec}]} = \frac{4\pi \left| AF\left(\theta_{[BS_{rec}]}, \phi_{[BS_{rec}]}\right)\right|^2 w\left(\theta_{[BS_{rec}]}, \phi_{[BS_{rec}]}\right)^2}{\int_0^{2\pi}\int_0^{\pi} |AF(\theta,\phi)|^2 w(\theta,\phi)^2 \sin\theta \mathrm{d}\theta \mathrm{d}\phi}\eta, \tag{3}$$

where $(\theta_{[BS_{rec}]}, \phi_{[BS_{rec}]})$ represents the direction towards the BS, $w(\theta, \phi)$ is the magnitude of the far-field beam pattern of each antenna element, and $\eta \in [0, 1]$ is the antenna array efficiency.

As for the interference model, we adopt the signal-to-interference-plus-noise ratio (SINR) to give the theoretical upper bounds on the channel capacity of the interfered communications. Accordingly, the SINR of the interfered BS for receiving data from a GU under the interference of the UVAA is as follows:

$$Y_{[BS_{in}]} = \frac{P^{rec}_{[GU]}}{\sigma^2 + P^{rec}_{[BS_{in}]}}, \tag{4}$$

where $P^{rec}_{[GU]}$ is the receiver power of the interfered BS from the GU which is set as a constant for the sake of simplicity in this work, and $P^{rec}_{[BS_{in}]}$ is the interference power from the UVAA.

**Propulsion Energy Consumption Model of UAV.** The expression for the propulsion energy consumption of a rotary-wing UAV in the two-dimensional (2D) horizontal plane can be modeled as follows [18]:

$$P(v) = \xi_1\left(1 + \frac{v^2}{\nu}\right) + \xi_2\left(\sqrt{1 + \frac{v^4}{\nu^2}} - \frac{v^2}{\nu}\right)^{1/2} + \xi_3 v^3, \tag{5}$$

where $v$ is the velocity of the UAV and other parameters can be seen as the constants related to the UAV, which be explained in detail in [17]. Moreover, the propulsion energy consumption of 3D UAV trajectory is expressed as follows [17]:

$$E(T) \approx \int_0^T P(v(t))dt + \frac{1}{2}m_U(v(T)^2 - v(0)^2) + mgh, \tag{6}$$

where $v(t)$ is the instantaneous UAV speed of time $t$. Moreover, $T$, $m_U$, $g$ and $h$ are the end time of the flight, aircraft mass of the UAV, gravitational acceleration and height changes, respectively.

### 2.2 Problem Formulation

The primary purpose of the considered CB-based communication system is to complete the data transmission mission as soon as possible, which can be achieved by improving the directivity towards the targeted BSs. Moreover, the interferences of the UVAA to the other BSs should be minimized to realize the interference mitigation. These two purposes above can be balanced by optimizing

the beam pattern of the UVAA, which means that the positions and excitation current weights should be designed carefully. However, due to the 3D movements, the large amount of propulsion energy of the UAVs will be consumed. Thus, the energy-efficient also should be considered during the flight design. Finally, the UVAA only needs to communicate with one BS, which means that the selection of a suitable receiver BS needs to be determined since it affects the transmission efficiency. Therefore, the aforementioned problems need to be jointly considered due to the existing trade-offs between them. Mathematically, the aforementioned optimization objectives can be expressed as follows.

***Optimization Objective 1:*** We first minimize the data transmission time of the considered system, which can be expressed as follows:

$$f_1(\boldsymbol{w}^U, \boldsymbol{x}^U, \boldsymbol{y}^U, \boldsymbol{z}^U, \boldsymbol{q}) = \frac{Data}{R_{[BS_{rec}]}}, \tag{7}$$

where $\boldsymbol{w}^U, \boldsymbol{x}^U, \boldsymbol{y}^U$ and $\boldsymbol{z}^U$ are the vectors of the excitation current weights, x-axis, y-axis and z-axis coordinates of the UAVs, respectively. Moreover, $\boldsymbol{q}$ is the ID of the selected BS and $Data$ is total amount of the collected data. Note that $\boldsymbol{X} = \{\boldsymbol{w}^U, \boldsymbol{x}^U, \boldsymbol{y}^U, \boldsymbol{z}^U, \boldsymbol{q}\}$ is the solution of the problem. Specifically, $\{\boldsymbol{w}^U, \boldsymbol{x}^U, \boldsymbol{y}^U, \boldsymbol{z}^U\}$ are continuous dimensions and $\boldsymbol{q}$ is the discrete dimension.

***Optimization Objective 2:*** We need to maximize the total SINRs of the interfered BSs for interference mitigation, which can be expressed as follows:

$$f_2(\boldsymbol{w}^U, \boldsymbol{x}^U, \boldsymbol{y}^U, \boldsymbol{z}^U, \boldsymbol{q}) = \sum_{n \in \{\mathcal{B} - \boldsymbol{q}\}} Y_{[BS_n]}, \tag{8}$$

where $Y_{[BS_n]}$ is the SINR of the $n$th BS.

***Optimization Objective 3:*** To increase the service time of UAVs, we minimize the energy consumption of the UAVs as follows:

$$f_3(\boldsymbol{x}^U, \boldsymbol{y}^U, \boldsymbol{z}^U, \boldsymbol{q}) = \sum_{m=1}^{N_{UAV}} E_m(T), \tag{9}$$

where $E_m(T)$ is the propulsion energy consumption of the $m$th UAV, and the corresponding calculation method can be found in [14]. Accordingly, the IMMOP consists of the three optimization objectives can be formulated as follows:

$$\begin{aligned}
(P1) \quad \min_{\boldsymbol{X}} \quad & \{f_1, -f_2, f_3\} && \text{(10a)}\\
\text{s.t.} \quad & 0 \leqslant \omega_m \leqslant 1, \forall m \in \mathcal{U} && \text{(10b)}\\
& (x_m^U, y_m^U, z_m^U) \in \mathbb{C}^3, \forall m \in \mathcal{U} && \text{(10c)}\\
& \boldsymbol{q} \in \{1, 2, ..., N_{BS}\} && \text{(10d)}\\
& D_{(m_1, m_2)} \geq D_{min}, \forall m_1, m_2 \in \mathcal{U} && \text{(10e)}
\end{aligned}$$

where $\mathbb{C}^3$ is the coordinate set of the 3D movable range of the UAVs. Moreover, the constraint in Eq. (10e) indicates that the minimum separation distance

between two adjacent UAVs must be greater than $D_{min}$ to avoid collision. Note that the formulated IMMOP is NP-hard since the second optimization objective can be simplified to be a nonlinear knapsack problem [3], and the proof is omitted due to the page limitation.

## 3 The Proposed Method

The IMMOP is NP-hard and sophisticated. Thus, we propose a CMOMVO method via the chaos theory to solve it in this part.

**Algorithm 1:** CMOMVO

**Input**: population size $N_{pop}$, maximum iteration $t_{max}$ and archive set *Archive*, etc.;
**Output**: Pareto solution set

```
for i = 1 to N_pop do
    Initialize the ith solution X_i of P by using Eq. (11);
for t = 1 to t_max do
    Calculate the objective function values of all solutions of P and update Archive;
    Update the algorithm parameters by using Eq. (12);
    for i = 1 to N_pop do
        Update the continues part of the ith solution (i.e., X_i(w^U, x^U, y^U, z^U)) by using method of conventional MOMVO;
        Update the discrete part of the ith solution (i.e., X_i(q)) via Eq. (13);
Return Archive;
```

### 3.1 CMOMVO

CMOMVO is extended and enhanced from the conventional MOMVO by introducing the concept of chaos theory, and the details are presented as follows.

**Solution Initialization.** We introduce the non-linear and semi-random of the chaos theory for improving the initial solution quality, i.e.,

$$\boldsymbol{X}_i = \boldsymbol{L_B} + \boldsymbol{C}_a^1 \times (\boldsymbol{U_B} - \boldsymbol{L_B}), \tag{11}$$

where $\boldsymbol{X}_i$ is the $i$th solution of the population $\boldsymbol{P}$. Moreover, $\boldsymbol{L_B}$ and $\boldsymbol{U_B}$ are the lower and upper bounds of a solution, and $\boldsymbol{C}_a^1$ is the chaotic array generated by the Logistic map [13].

**Algorithm Parameter Update.** We map the chaotic array into the algorithm parameter space. Thus, TDR and WEP are updated as follows:

$$\mathrm{TDR}^t = \mathrm{TDR}^t \times \boldsymbol{C}_a^2(t), \quad \mathrm{WEP}^t = \mathrm{WEP}^t \times \boldsymbol{C}_a^3(t), \tag{12}$$

where $\mathrm{TDR}^t$ and $\mathrm{WEP}^t$ are the values of TDR and WEP in $t$th iteration, respectively. Moreover, $\boldsymbol{C}_a^2(t)$ and $\boldsymbol{C}_a^3(t)$ are the $t$th element of the Sine and Circle maps [13], respectively.

**Solution Update.** MOMVO cannot handle the discrete dimension $q$. Thus, we propose a chaos-based discrete solution update method as follows:

$$\boldsymbol{q}_{t+1,i} = \begin{cases} \boldsymbol{q}_t^{best} & \boldsymbol{C}_a^2(t) > \mathrm{rand} \\ \boldsymbol{q}_{t,i} & \text{otherwise} \end{cases}, \tag{13}$$

where $\boldsymbol{q}_{t,i}$ is the discrete dimension of the $i$th solution in the $t$th iteration, $\boldsymbol{q}_t^{best}$ is the discrete dimension of the best solution (defined by MOMVO), and rand is a random value.

### 3.2 Main Steps and Analysis of CMOMVO

The pseudo-code of the proposed CMOMVO is shown in Algorithm 1. Note that *Archive* is a set for storing the optimal solutions since the multi-objective optimization can obtain several Pareto optimal solutions (PSs) [9]. Moreover, the complexity of the proposed CMOMVO is $\mathcal{O}(N_{obj} \cdot N_{pop}^2)$ when $N_{obj}$ represents the number of optimization objectives.

## 4 Simulation Results and Analysis

In our simulation, the pathloss exponent, transmit power of a UAV, transmit power of a GU and carrier frequency are set as 3, 0.1 W, 2 W and 2.4 GHz, respectively. Moreover, the number of the UAVs and BSs are set as 8 and 8, respectively. Other key parameters follow [12,18]. In addition, the multi-objective ant lion optimizer (MOALO) [10], multi-objective dragonfly algorithm (MODA) [8], multi-objective grasshopper optimization algorithm (MOGOA) [11] and MOMVO are introduced as the benchmarks.

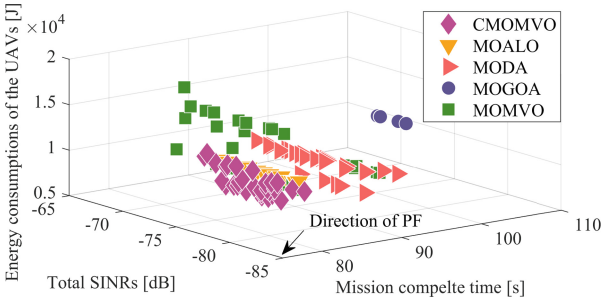



**Fig. 2.** Solution distributions of different algorithms.

**Table 1.** Numerical results obtained by different algorithms.

| Method | $f_1$ [s] | $f_2$ [dB] | $f_3$ [J] |
|---|---|---|---|
| CMOMVO | **82.91** | **81.13** | $\mathbf{9.80 \times 10^3}$ |
| MOALO | 84.78 | 74.48 | $9.85 \times 10^3$ |
| MODA | 90.60 | 73.10 | $1.03 \times 10^4$ |
| MOGOA | 99.06 | 75.93 | $1.39 \times 10^4$ |
| MOMVO | 83.69 | 70.02 | $1.74 \times 10^4$ |

Table 1 shows the numerical results in terms of data transmission time ($f_1$), total SINRs ($f_2$) and total propulsion energy consumptions ($f_3$) obtained by different algorithms. As can be seen, the other BSs can obtain sufficient SINR for communication. Moreover, the proposed CMOMVO achieves the best results on all the optimization objectives. In addition, the solution distributions obtained by different algorithms are given in Fig. 2, which shows that the solutions obtained by CMOMVO are much closer to the true Pareto front (PF) [9] and have a more uniform distribution. Thus, the proposed CMOMVO has the best performance among all benchmarks. The reason may be that the introduced chaos-based methods can balance the exploitation and exploration abilities of the algorithm, thereby increasing the search efficiency. In summary, the proposed method can solve the IMMOP efficiently and outperforms other benchmarks.

## 5 Conclusion

In this work, we propose a novel CB-based interference mitigation method of the UAV-enabled data collection scenario and formulate an IMMOP which can simultaneously reduce the data transmission time, improve the SINRs of the terrestrial networks and reduce the propulsion energy consumptions of the UAVs. Moreover, a CMOMVO with several improved factors is proposed for solving the IMMOP. Simulation results demonstrate that the proposed CMOMVO outperforms MOALO, MODA, MOGOA and MOMVO.

**Acknowledgment.** This study is supported in part by the National Natural Science Foundation of China (62172186, 62002133, 61872158), in part by the National Key Research and Development Program of China (2018YFC0831706), in part by the Science and Technology Development Plan Project of Jilin Province (20210101183JC, 20210201072GX, 20200201166JC), in part by the Young Science and Technology Talent Lift Project of Jilin Province (QT202013), and in part by the Central Government funds for guiding local scientific and Technological Development (2021Szvup047).

## References


1. Balanis, C.A.: Antenna Theory: Analysis and Design. John Wiley, Hoboken, NJ (2005)
2. Li, J., Kang, H., Sun, G., Liang, S., Liu, Y., Zhang, Y.: Physical layer secure communications based on collaborative beamforming for UAV networks: a multi-objective optimization approach. In: Proceedings of the IEEE INFOCOM, pp. 1–10 (2021)
3. Liang, S., et al.: A joint optimization approach for distributed collaborative beamforming in mobile wireless sensor networks. Ad Hoc Netw. **106**, 102216 (2020)
4. Lin, C., Wang, Z., Deng, J., Wang, L., Ren, J., Wu, G.: mTS: temporal-and spatial-collaborative charging for wireless rechargeable sensor networks with multiple vehicles. In: Proceedings of the IEEEINFOCOM, pp. 99–107. IEEE (2018)
5. Lin, C., Zhou, J., Guo, C., Song, H., Wu, G., Obaidat, M.S.: TSCA: a temporal-spatial real-time charging scheduling algorithm for on-demand architecture in wireless rechargeable sensor networks. IEEE Trans. Mob. Comput. **17**(1), 211–224 (2018)
6. Liu, T., Cui, M., Zhang, G., Wu, Q., Chu, X., Zhang, J.: 3D trajectory and transmit power optimization for UAV-enabled multi-link relaying systems. IEEE Trans. Green Commun. Netw. **5**(1), 392–405 (2021)
7. Mei, W., Wu, Q., Zhang, R.: Cellular-connected UAV: uplink association, power control and interference coordination. IEEE Trans. Wirel. Commun. **18**(11), 5380–5393 (2019)
8. Mirjalili, S.: Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput. Appl. **27**(4), 1053–1073 (2016)
9. Mirjalili, S., Jangir, P., Mirjalili, S.Z., Saremi, S., Trivedi, I.N.: Optimization of problems with multiple objectives using the multi-verse optimization algorithm. Knowl. Based Syst. **134**, 50–71 (2017)
10. Mirjalili, S., Jangir, P., Saremi, S.: Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. Appl. Intell. **46**(1), 79–95 (2017)
11. Mirjalili, S.Z., Mirjalili, S., Saremi, S., Faris, H., Aljarah, I.: Grasshopper optimization algorithm for multi-objective optimization problems. Appl. Intell. **48**(4), 805–820 (2017). https://doi.org/10.1007/s10489-017-1019-8
12. Mozaffari, M., Saad, W., Bennis, M., Debbah, M.: Communications and control for wireless drone-based antenna array. IEEE Trans. Commun. **67**(1), 820–834 (2019)
13. Sayed, G.I., Tharwat, A., Hassanien, A.E.: Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection. Appl. Intell. **49**(1), 188–205 (2019)
14. Sun, G., Li, J., Liu, Y., Liang, S., Kang, H.: Time and energy minimization communications based on collaborative beamforming for UAV networks: a multi-objective optimization method. IEEE J. Sel. Areas Commun. **39**(11), 3555–3572 (2021)
15. Wang, Z., Duan, L.: Chase or wait: dynamic UAV deployment to learn and catch time-varying user activities. IEEE Trans. Mob. Comput. (2021). https://doi.org/10.1109/TMC.2021.3107027
16. Wu, Q., et al.: A comprehensive overview on 5G-and-beyond networks with UAVs: from communications to sensing and intelligence. IEEE J. Sel. Areas Commun. **39**(10), 2912–2945 (2021)

17. Zeng, Y., Wu, Q., Zhang, R.: Accessing from the sky: a tutorial on UAV communications for 5G and beyond. Proc. IEEE **107**(12), 2327–2375 (2019)
18. Zeng, Y., Xu, J., Zhang, R.: Energy minimization for wireless communication with rotary-wing UAV. IEEE Trans. Wirel. Commun. **18**(4), 2329–2345 (2019)
19. Zhang, X., Duan, L.: Energy-saving deployment algorithms of UAV swarm for sustainable wireless coverage. IEEE Trans. Veh. Technol. **69**(9), 10320–10335 (2020)

# Authentication Mechanism Based on Physical Layer Security in Industrial Wireless Sensor Networks

Ruizhong Du[1,2], Lin Zhen[1(✉)], and Yan Liu[3]

[1] School of Cyberspace Security and Computer Science, Hebei University, Baoding, China
durz@hbu.edu.cn, snowlin_zhen@163.com

[2] Hebei Key Laboratory of Highly Trusted Information System, Hebei University, Baoding, China

[3] School of Systems Information Science, Future University Hakodate, Hakodate, Japan
g3220002@fun.ac.jp

**Abstract.** When wireless network technology is applied to industrial scenes, the open channel environment makes industrial equipment more vulnerable to attacks and threats from illegal nodes, such as eavesdropping, deception and identity information forgery. The complexity and variety of attack methods make the supervised machine learning algorithm insufficient to establish a low complexity, lightweight and high security authentication mechanism in industrial wireless sensor networks. Moreover, the wireless electromagnetic wave will be affected by additive noise and fading in the propagation process, making the wireless channel in a dynamic state. Based on this, we study a new authentication mechanism based on physical layer security for wireless sensor networks in dynamic industrial scenarios. Using more precise physical layer channel information, and building an authentication model around positive-unlabeled (PU) learning and bootstrap aggregating (bagging) strategy, we can accurately distinguish legal nodes and illegal nodes in the received channel information in the industrial scene where only the channel information of legal nodes is known. Finally, the effectiveness of the scheme is verified by using the public data set collected by the national institute of standards and technology (NIST) in a real industrial scene.

**Keywords:** Industrial wireless sensor network · Physical layer security · Machine learning


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 567–578, 2022.
https://doi.org/10.1007/978-3-031-19208-1_47

## 1 Introduction

When wireless network technology is applied to industrial scenarios, the open channel environment makes industrial equipment more vulnerable to attack threats from illegal nodes, such as eavesdropping, spoofing, and identity information forgery. In addition, wireless communication terminals have limited computing power, and traditional cryptography-based security encryption mechanisms are difficult to execute on the terminals [1–3]. Therefore, the main problem faced by wireless sensor networks in industrial scenarios is to find a lightweight and high-security authentication scheme. The authentication research based on physical layer security [4–7] solves the dilemma of limited resources and high security requirements in a mobile communication system with the characteristics of lightweight and high reliability. Therefore, this paper proposes a new authentication scheme based on physical layer security for wireless sensor networks in industrial scenarios to detect nodes from illegal sources.

Our first contribution is to consider a more reasonable intrusion scenario of illegal nodes. Assuming that the physical layer channel information of all legal nodes is known and the physical channel information of illegal nodes is unknown, an authentication model based on PU bagging strategy is constructed; The second contribution is to use more refined physical layer channel information to transform the high-dimensional channel information data set into a low-dimensional feature set, so as to improve the authentication performance and efficiency of the model. Finally, the authentication strategy is applied to the public data set in the real industrial scene, and the simulation results verify the effectiveness of the scheme.

## 2 Related Work

Considering that industrial networks are vulnerable to intrusion and attacks, researchers have also developed many effective authentication schemes. Liao et al. [8] proposed a physical layer authentication scheme based on multiuser combined with deep neural network algorithm. The effectiveness of the scheme is verified in static and dynamic industrial scenarios; F. Pan et al. [9] transformed the physical layer authentication process of distinguishing legitimate and attacker into a binary classification process, and proposed a threshold free physical layer authentication method based on machine learning to learn and classify the channel state information (CSI) of legitimate and illegal nodes; in order to improve the accuracy of model detection, F. Pan et al. [10] used the physical layer reputation obtained by channel state information accumulation and back propagation neural network for clone detection; S. L. Chen et al. [11] used the difference of physical layer channel information between legal nodes and illegal nodes to propose a physical layer authentication scheme based on unsupervised and supervised learning to detect clone attack and sybil attack at the same time; Marabissi et al. [7] used classification and regression tree (CART) algorithm and random forest algorithm to propose a scheme of Internet of things node authentication and identity spoofing detection based on physical layer security.

The above authentication schemes based on physical layer security assume that the attacker's physical channel information is known, that is, the authentication model is constructed by using supervised machine learning algorithm. However, data exchange in industrial wireless sensor networks mostly requires real-time transmission, and the attack methods of illegal nodes are complex and changeable. It is obviously difficult to record the physical layer channel information of illegal nodes detected each time for subsequent authentication. Therefore, only using supervised machine learning algorithm to build authentication model is not enough for practical industrial scene applications. In order to overcome this shortcoming, we propose a physical layer authentication strategy based on PU bagging algorithm. This scheme is more suitable for industrial scenarios that do not require high types of illegal nodes and attack methods.

## 3 Authentication Mechanism Based on Physical Layer Security

In the framework shown in Fig. 1, this paper assumes that only the channel information of legitimate nodes is known, while the channel information of illegal nodes is unknown. That is, the system already knows who the legal node is. This is because when the terminal node accesses the communication system for the first time, it will first carry out the traditional upper layer authentication to obtain the initial physical layer channel information. After the first upper layer authentication is successful, the system extracts the physical layer channel information from the initial information packet and pastes corresponding labels (for example, node 1, node 2,..., node $n$) to carry out the physical layer authentication of subsequent information packets.

When a new node accesses the communication system, if you want to judge whether the node belongs to a legal node or an illegal node, you only need to identify and authenticate the channel information in the physical layer. The specific process is as follows: firstly, we extract the amplitude, phase, carrier frequency offset (i.e. phase change) and variance features of physical layer channel information to form a feature set representing channel state information and verify its importance; then, under the condition that the physical layer channel information of the legal node is known, a physical layer authentication scheme based on PU bagging is constructed to distinguish the legal node and the illegal node in the newly received channel information.

### 3.1 Construction of Feature Set of Channel State Information

The difference of channel state information in amplitude, phase and carrier frequency offset information is the key to distinguish legal nodes from illegal nodes. Carrier frequency offset information is the change of phase. These characteristics can enhance the recognition and classification effect of authentication process [12–14]. In this paper, the mean and variance of amplitude information, phase

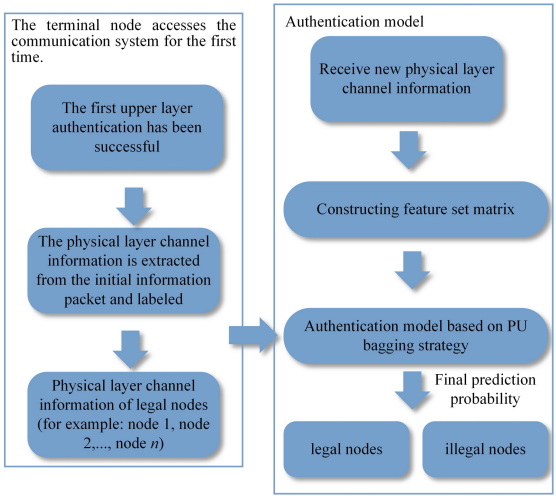



**Fig. 1.** Physical layer authentication framework based on integrated PU learning.

information and carrier frequency offset information of channel state information are extracted, and the feature set is constructed by combining their variance characteristics. The details are as follows:

In the industrial scenario, the physical layer channel state information is collected by the legal receiver, and its signal model can be expressed as follows [11]:

$$r(t) = Hx(t) + n(t), \tag{1}$$

where $t$ represents the time slot, that is, the time interval between each data frame; $r(t)$ is the signal vector at the receiving end; $x(t)$ is the signal vector at the transmitting end; $H$ represents the channel state matrix; $n(t)$ stands for Gaussian white noise. The channel state information is the channel state matrix $H$, which is the set of channel information of each subcarrier:

$$H = [H_1, H_2, ..., H_n, ..., H_N]^{\mathrm{T}}, \tag{2}$$

where each $H_n$ represents a subcarrier, $n = 1, 2, ..., N$; $N$ is the number of subcarriers contained in each channel state information. Each subcarrier in the channel state matrix $H$ appears in the complex form $a(t) + b(t) * i$.

The instantaneous amplitude and instantaneous phase of the channel state information can be calculated as follows:

$$A(t) = [a^2(t) + b^2(t)]^{\frac{1}{2}}, \tag{3}$$

$$\theta(t) = tan^{-1}[\frac{a(t)}{b(t)}], \tag{4}$$

where $a(t)$ and $b(t)$ represent the in-phase and quadrature components of the complex vector. The carrier frequency offset information is the change of phase, which can be calculated as follows:

$$\omega(t) = \frac{1}{2\pi}\frac{d\theta(t)}{dt}. \tag{5}$$

The physical layer channel state information data set [17] collected by NIST are $8188 \times 1$ dimensional complex vectors. Therefore, this paper maps high-dimensional features to two-dimensional features by calculating the mean and variance of each complex vector feature. Taking the instantaneous amplitude as an example, note that the amplitude of the $n^{th}$ frame transmitted by the $k^{th}$ node in the acquisition environment is $A_k(n)$, where $n = 1, 2, ..., N$. Calculate the mean and variance of the amplitude samples collected at the $k^{th}$ node:

$$Mean(A_k) = \frac{1}{N}\sum_{n=1}^{N} A_k(n). \tag{6}$$

$$Var(A_k) = \frac{1}{N}\sum_{n=1}^{N}[A_k(n) - Mean(A_k)]^2. \tag{7}$$

Similarly, note that the phase and carrier frequency offset information of the $n^{th}$ frame transmitted by the $k^{th}$ node in the acquisition environment are $\theta_k(n)$ and $\omega_k(n)$ respectively, where $n = 1, 2, ..., N$.

Owing to the influence of random noise and other factors, the collected physical layer channel information usually fluctuates randomly in a certain range. The variance feature can describe the fluctuation range of channel state information [15,16], which can further enhance the authentication effect of legal nodes and illegal nodes. It can be calculated as follows:

$$\sigma^2 = \frac{1}{N-1}\sum_{n=1}^{N} |H_n - H_\mu|^2, \tag{8}$$

where $H_n$ represents the subcarrier data of a node in the $n^{th}$ frame, and $H_\mu$ represents the average value of subcarrier data of a node. To sum up, this paper extracts the amplitude information, phase information, carrier frequency offset information (i.e. phase change) and variance characteristics of channel state information to construct a feature set with dimension 7, as shown below:

$$F_k' = <Mean(A_k), Var(A_k), Mean(\theta_k), Var(\theta_k), Mean(\omega_k), Var(\omega_k), \sigma^2>. \tag{9}$$

The importance of the extracted channel state information features in an automotive assembly factory environment is shown in Fig. 2. From the effective value distribution of features, the channel state information features extracted in this paper behave differently under this dataset, but they are all important. The subsequent simulation results also verify the effectiveness of the feature set.

### 3.2 Authentication Model Based on PU Bagging Strategy

In a real industrial scenario, the number of illegal nodes is far less than that of legal nodes. Considering this situation, we use bagging algorithm as the framework to build a physical layer authentication scheme based on PU learning. According to the experimental verification, the PU bagging algorithm is more suitable for

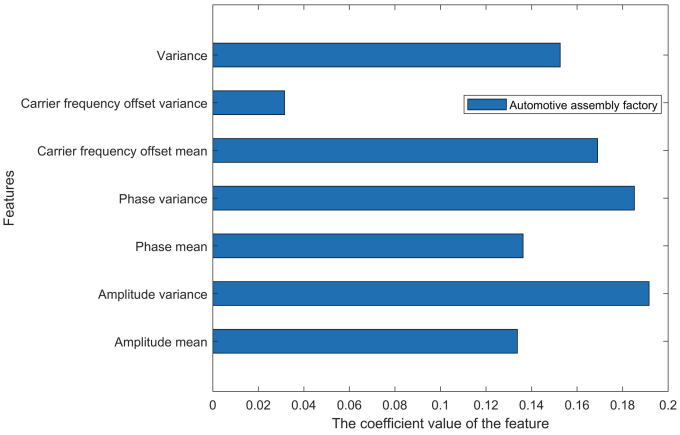



**Fig. 2.** Feature importance in Automotive Assembly Factory.

this data distribution and inherits the advantages of the bagging method. When the unlabeled sample set is large, the data processing speed can be accelerated through parallel operation, so as to improve the authentication efficiency. Firstly, we transform the authentication process of legal nodes and illegal nodes into a binary classification process. Then, PU bagging strategy is used to learn and classify the channel state information, so as to accurately distinguish legal nodes from illegal nodes. The specific authentication process is shown in Algorithm 1. Where $P$ represents the sample set of physical layer channel state information of known legal nodes, which is recorded as a positive sample; $U$ represents the unlabeled physical layer channel state information sample set, which is recorded as a negative sample; $M$ is the number of hypothetical prediction samples; $NU$ is a subset of negative samples with the same number of elements as $P$.

**Algorithm 1**: Authentication Model

**input**: $P$, $U$.
**output**: Average prediction probability of $T$ weak classifiers.
1. Constructing a seven-dimensional feature set:
$F'_k =< Mean(A_k), Var(A_k), Mean(\theta_k), Var(\theta_k), Mean(\omega_k), Var(\omega_k), \sigma^2 >$;
2. Use bootstrap method to randomly and repeatedly select $NU$ from $U$ and iterate $T$ times;
3. $P$ and $NU$ are trained together as training sets to obtain $T$ weak classifiers;
4. The prediction probability matrix of each weak classifier is obtained: $\begin{bmatrix} P_{l1} & P_{i1} \\ P_{l2} & P_{i2} \\ \vdots & \vdots \\ P_{lm} & P_{im} \end{bmatrix}$, $m = 1, 2, ..., M$;
5. Each iteration repeats the process of sampling, training and prediction;
6. The average prediction probability matrix of $T$ weak classifiers is obtained: $\begin{bmatrix} \overline{P}_{l1} & \overline{P}_{i1} \\ \overline{P}_{l2} & \overline{P}_{i2} \\ \vdots & \vdots \\ \overline{P}_{lm} & \overline{P}_{im} \end{bmatrix}$;
7. When $\overline{P}_{lm} > \overline{P}_{im}$, the unmarked channel information belongs to a legal node;
when $\overline{P}_{lm} < \overline{P}_{im}$, the unmarked channel information belongs to an illegal node;
8. Authentication complete.

## 4 Simulation Results and Analysis

This paper mainly studies the physical layer authentication of wireless communication system in dynamic industrial scene, that is, the authentication scheme of wireless channel in dynamic situation. We use the channel state information dataset collected by NIST under an automobile assembly factory of the real industrial site to conduct the scheme feasibility study [17]. There are many obstructions in the environment of automobile assembly factory, resulting in no line of sight component from the transmitter to the receiver, which is a typical Rayleigh fading channel. Rayleigh fading is a special kind of multipath fading, so the channel information data set collected by NIST meets the requirements of wireless channel in dynamic condition [18]. In this scenario, a total of 106 channel measurement positions were measured, and each measurement position had 300 records. Due to the unexpected pause and slow movement of the transmitter during the data measurement, 60 positions are selected as the effective node positions. The moving speed was slightly different in different sections, and the average moving speed was 4.4 mm/s.

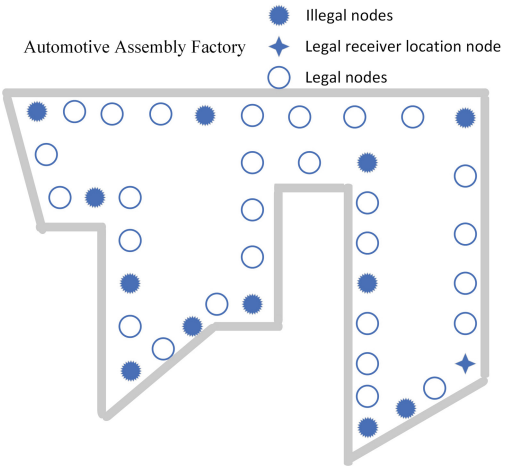



**Fig. 3.** Simulation experiment of illegal node intrusion under automobile assembly.

As shown in Fig. 3, in order to meet the condition that the number of illegal nodes accounts for different proportions of all nodes, we assumed that a fixed channel measurement location was a legal receiver node and other locations were legal nodes and illegal nodes deployed according to the corresponding proportion. For example, 60 nodes are deployed in the network. According to the proportion of illegal nodes accounting for 20% and 40% of all nodes, the number of illegal nodes is 12 and 24 respectively. By changing the proportion $P\_ratio = 0.1, 0.2, 0.3, 0.4, 0.5$ of known label legitimate node sample set, the classification effect of PU bagging strategy is investigated. In order to get the

best performance of the prediction model in the authentication scheme, our data are recorded under the optimal threshold. The optimal threshold *thr_best* refers to the threshold corresponding to the maximum difference between the true positive rate (TPR) and the false positive rate (FPR). Under the optimal threshold, the model can distinguish legal nodes from illegal nodes to the greatest extent; the predicted performance is the best; the Accuracy value is also the highest.

**Simulation of Different Proportions of Illegal Nodes.** Next, we verified the authentication performance of the PU bagging strategy under *Ill_nodes_ratio*. *Ill_nodes_ratio* represents the proportion of illegal nodes in all nodes. Figure 4 intuitively show the Accuracy and AUC values of the authentication model under different proportions of illegal nodes. The test data show that the Accuracy and AUC value of the authentication model improved with the increase of the proportion of CSI sample set of legal nodes with known labels, and the difference interval is [0.009, 0.019]; with the increase of the proportion of illegal nodes, the Accuracy and AUC values predicted by the model gradually increase, and the growth range is [0.009,0.023]. The simulation results show that in the industrial scene with a small number of illegal nodes, the authentication strategy proposed in this paper can accurately distinguish between legal nodes and illegal nodes.

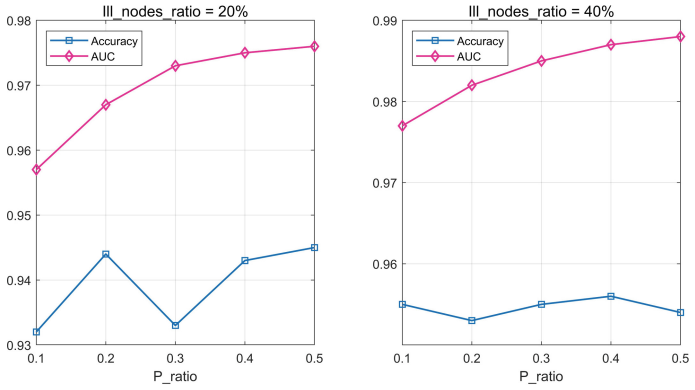



**Fig. 4.** Accuracy, TPR, FPR, AUC at *Ill_nodes_ratio* = 20%, 40%.

**Comparative Experiment.** We compare the Accuracy, AUC and efficiency of the model when three channel matrices with different dimensions are used as input respectively. The original dimension of the channel state information is 8188 dimensions. In order to reduce the amount of computation: in [8], the matrix dimension obtained by downsampling is 1638 dimensions; in reference [18], the optimal channel matrix dimension is 328 dimensions through downsampling processing. As shown in Fig. 5, the Accuracy difference of three channel

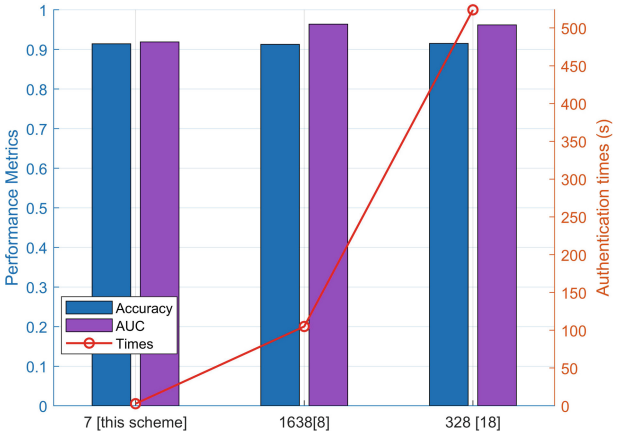



**Fig. 5.** Model prediction results of channel matrices with different dimensions.

state information matrices with different dimensions is very small; the AUC difference is distributed between the interval [0.053, 0.055]. However, in terms of authentication time, the scheme in this paper can distinguish legal nodes from illegal nodes in 60 nodes in only 2.8s; when using the channel matrix dimension processed by downsampling in [8,18] as the input, it takes 105s and 524s respectively to complete an authentication process. This shows that the authentication scheme proposed in this paper achieves better authentication performance and efficiency at the same time.

To better verify the authentication strategy proposed in this paper, a comparative experiment is carried out with the physical layer authentication scheme in [11]. Comparing the authentication performance of the two schemes to distinguish between legal nodes and illegal nodes, both are simulated to deploy 60 nodes in the network, and the illegal nodes account for 20% of all nodes, that is, under the same condition that the number of illegal nodes was 12. As shown in Fig. 6 and Table 1, we can observe that the authentication strategy proposed in this paper is close to the authentication performance of the scheme in [11] in terms of the distinction between legal nodes, illegal nodes and clone nodes. Under the condition of low FPR, the TPR value on the red and blue curves can reach 1, indicating the effectiveness of detecting illegal nodes and clone nodes; compared with the performance of distinguishing legal nodes and sybil nodes in [11], the scheme in this paper can distinguish legal nodes and illegal nodes more accurately. To sum up, the authentication scheme proposed in this paper does not pay attention to distinguish which attack type the illegal node belongs to. It is obviously more universal and more suitable for user security management in practical industrial wireless sensor networks. At the same time, compared with the supervised learning model, PU learning can reduce the requirements of training data. Unknown illegal nodes in the physical layer channel information can be mined only through the legal node data, which effectively reduces the collection of physical layer channel information of illegal nodes.

**Table 1.** Comparison of model prediction results

| | Threshold | AUC | Accuracy |
|---|---|---|---|
| Legal nodes and illegal nodes | 0.3868 | 0.976 | 0.945 |
| Legal nodes and clone nodes [11] | 0.7 | 0.998 | 1 |
| Legal nodes and sybil nodes [11] | 0.9 | 0.789 | 1 |

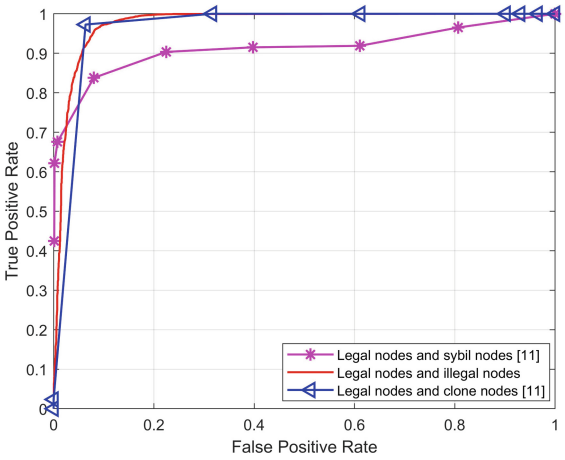



**Fig. 6.** Comparison diagram of simulation experiment results.

## 5 Conclusion

This paper proposes a PU bagging authentication scheme based on physical layer security for industrial wireless sensor networks. This strategy does not need to collect and record a large number of illegal node channel information in advance, which saves some overhead, and can accurately distinguish legal nodes and illegal nodes in the received channel information when only the channel information of the legal node is known. In order to improve the authentication efficiency, this paper extracts the amplitude, phase, carrier frequency offset (i.e. the change of phase) and variance of channel state information, constructs a feature set and verifies its feature importance. Thus, it solves the limitation of many machine learning problems owing to the direct use of high-dimensional CSI data for simulation experiments. Finally, the effectiveness of the scheme is verified by using the data set in the real industrial environment. In future research, we will study other characteristics of physical layer channel information and try to use other machine learning schemes to further improve the authentication performance and enhance the universality of the scheme.

**Acknowledgements.** This project received funding from the Natural Science Foundation of China under Grant 61170254, and from the Hebei Natural Science Foundation Key Program Project under Grant F2019201290.

## References

1. Islam, S.-N., Baig, Z., Zeadally, S.: Physical layer security for the smart grid: vulnerabilities, threats, and countermeasures. IEEE Trans. Industr. Inform. **15**(12), 6522–6530 (2019)
2. Bhamare, D., Zolanvari, M., Erbad, A., et al.: Cybersecurity for industrial control systems: a survey. Comput. Secur. **89**, 101677 (2020)
3. Zhang, P., Shen, Y., Jiang, X., et al.: Physical layer authentication jointly utilizing channel and phase noise in MIMO systems. IEEE Trans. Commun. **68**, 2446–2458 (2020)
4. Yadav, N., Pande, S., Khamparia, A., Gupta, D.: Intrusion detection system on IoT with 5G network using deep learning. Wirel. Commun. Mob. Comput. **2022**, 9304689 (2022)
5. Bashar, A., Smys, S.: Physical layer protection against sensor eavesdropper channels in wireless sensor networks. IRO J. Sustain. Wirel. Syst. **3**(2), 59–67 (2021)
6. Gao, N., Ni, Q., Feng, D.: Physical layer authentication under intelligent spoofing in wireless sensor networks. Signal Process. **166**, 107272 (2020)
7. Marabissi, D., Mucchi, L., Stomaci, A.: IoT nodes authentication and ID spoofing detection based on joint use of physical layer security and machine learning. Future Internet **14**(2), 61 (2022)
8. Liao, R.-F., Wen, H., Chen, S.-L., et al.: Multiuser physical layer authentication in internet of things with data augmentation. IEEE Internet Things J. **7**(3), 2077–2088 (2020)
9. Pan, F., Pang, Z.-B., Wen, H., et al.: Threshold-free physical layer authentication based on machine learning for industrial wireless CPS. IEEE Trans. Industr. Inf. **15**, 6481–6491 (2019)
10. Pan, F., Wen, H., Gao, X., et al.: Clone detection based on BPNN and physical layer reputation for industrial wireless CPS. IEEE Trans. Industr. Inf. **17**, 3693–3702 (2020)
11. Chen, S.-L., Pang, Z.-B., Wen, H., et al.: Automated labeling and learning for physical layer authentication against clone node and Sybil attacks in industrial wireless edge networks. IEEE Trans. Industr. Inf. **17**(3), 2041–2051 (2020)
12. Xie, F., Wen, H., Li, Y., et al.: Optimized coherent integration-based radio frequency fingerprinting in internet of things. IEEE Internet Things J. **5**, 3967–3977 (2018)
13. Lee, W., Lee, K.: Deep learning-aided distributed transmit power control for underlay cognitive radio network. IEEE Trans. Veh. Technol. **70**, 3990–3994 (2021)
14. Wu, B., Qiu, W., Jia, J., et al.: Landslide susceptibility modeling using bagging-based positive-unlabeled learning. IEEE Geosci. Remote Sens. Lett. **18**, 766–770 (2020)
15. Bekker, J., Davis, J.: Learning from positive and unlabeled data: a survey. Mach. Learn. **109**, 719–760 (2020)
16. Zhao, J., Liu, N.: A safe semi-supervised classification algorithm using multiple classifiers ensemble. Neural Process. Lett. **53**(4), 2603–2616 (2020). https://doi.org/10.1007/s11063-020-10191-1

17. Candell, R., Remley, K.-A., Moayeri, N.: Radio frequency measurements for selected manufacturing and industrial environments. NIST, Tech. Rep. 1951 (2016). http://doi.org/10.18434/T44S3N
18. Du, R.-Z., Zhen, L.: Multiuser physical layer security mechanism in the wireless communication system of the IIOT. Comput. Secur. **113**, 102559 (2022)

# A Practical Data Authentication Scheme for Unattended Wireless Sensor Networks Using Physically Unclonable Functions

Pingchuan Wang[1], Lupeng Zhang[1], Jinhao Pan[1], and Fengqi Li[2(✉)]

[1] Dalian University of Technology, Dalian, China
[2] Dalian Jiaotong University, Dalian, China
Fengqi-Li@outlook.com

**Abstract.** In Unattended Wireless Sensor Networks (UWSNs), an itinerant sink does not establish a continuous real-time channel with sensors. The sensed data needs to be temporarily stored in the off-line nodes. Due to the unattended nature of sensors, adversaries can easily compromise the sensors and tamper with data by physical method. Therefore, it is necessary to design a data authentication scheme to identify the tampered data and against physical attacks. Existing research mostly relies on a trusted long-term third party or cooperation mechanism, which makes the scheme difficult to be implemented. In this paper, we introduce Physically Unclonable Functions (PUF) to design a practical data authentication scheme for UWSNs. In the scheme, we first design an authentication and key agreement (AKA) protocol using challenge-response pairs (CRPs). After establishing a symmetric channel, we propose a PUF-based Message Authentication Code (MAC) scheme to ensure data security from the source. We also give a security analysis and a practical implementation of PUF. The result shows the implementable and security of our scheme.

**Keywords:** Unattended Wireless Sensor Networks · Physically unclonable functions · Data authentication

## 1 Introduction

Wireless Sensor Networks (WSNs) have been widely used in environmental monitoring and data collection in recent years. Traditional WSNs establish the long-term real-time channel between sensors and sinks by multi-hop routing protocol. But considering some extreme scenarios (battlefield, underwater environment, forest, etc.), it is difficult to pre-plan a complex multi-hop network [1]. It would be convenient to deploy a network in which sensors are distributed in an impromptu manner and no multi-hop communication is created, which is named the Unattended Wireless Sensor Networks (UWSNs).

In UWSNs, the data collection model changes from real-time to offline. The itinerant sinks move among sensors and present in the network periodically.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 579–590, 2022.
https://doi.org/10.1007/978-3-031-19208-1_48

Sensors have to wait until the sink is available so that they can upload their sensed data. Therefore, the sensed data must be temporarily stored in the sensor nodes. In addition, unattended sensors are usually mass-produced devices with no secure hardware or tamper-resistance components, and are frequently deployed in hostile environments. The above properties of UWSNs introduce a vulnerability period in which a mobile adversary (ADV from now on) can compromise nodes and alter or erase sensitive data that has not been uploaded to sinks [2]. Therefore, a data authentication scheme is needed to ensure the integrity and availability of sensed data.

Extensive studies have focused on the subject of data authentication for UWSNs. Traditional encryption schemes use Message Authentication Code (MAC) to ensure the reliability of data. But unattended sensors face the risk of key exposure [3]: ADV can physically hack the circuits to cause a memory dump. [4] presents more discussion about the application of encryption in UWSNs. To propose a reasonable data authentication scheme, previous research generally made two kinds of compromises on the network model: long-term connection or multi-hop route. At the long-term connection compromise, several research describe sink-to-sensors broadcast authentication schemes [5,6]. But they assume a constantly present sink, so they are not suitable for applying to UWSNs. At the multi-hop route compromise, the schemes assume that any two sensors can communicate either through a multi-hop network. For example, [7,8] propose an authentication scheme based on a list of MAC-s which indicate the route towards the sink and will be verified recursively along the route. Other studies focus on collaborative authentication scheme [2,9]. They achieve data authentication based on sensor co-operation. However, it is not easy to establish a multi-hop network in the complex environment of UWSNs, which will cause a relatively high communication complexity. Therefore, a more practical scheme is needed to ensure data reliability and against physical attacks in UWSNs.

Physically Unclonable Function (PUF) is believed to play an important role in protection against physical attacks. It was first proposed as a hardware feature of CMOS caused by physical random nanoscale disarray phenomenon [10,11]. This phenomenon can be used to generate keys without having to store any sensitive information in devices' memory [12,13]. The key generated in this way is unclonable and anti-physical-attack, any manipulation of circuit will destroy the secret. Several schemes have been proposed for device authentication and key establishment using strong PUF in the past [14,15]. To the best of our knowledge, we are the first to consider in UWSNs using PUF for data authentication.

In this paper, we propose a practical data authentication scheme for UWSNs based on a lightweight PUF implementation. First, we analyze the UWSNs network model and focus on the sensor-to-sink data authentication process. Second, we ensure the premise of data reliability, which is authenticating sensors' identity and establishing a secret key between sensors and sinks. Only if this achieved, the security mechanism can protect data both in the transmission period and the temporary storage period. Next, we design a round-by-round MAC scheme running at the data source to resist the physical attack from ADV. We also

provide a brief security analysis and a practical implementation of PUF, which proves the practicality and security of our scheme. The main contributions of this paper are as follows:

1. We propose a lightweight authentication and key agreement protocol based on PUF (PUF-AKA) for both forward and backward security, relying on challenge-response pairs (CRPs) to generate a secret key that does not need to be stored in the memory.
2. We propose a PUF-based MAC scheme (PUF-MAC) to protect data from altering by ADV. We use the symmetric key and CRPs of PUF to perform round-by-round MAC and encryption operations.
3. We conduct a security analysis of our scheme. The result shows that our scheme has the weakest assumption and the best security performance.

The remainder of the paper is organized as follows. In Sect. 2, we introduce the system model and the network assumptions. In Sect. 3, we propose the practical data authentication scheme, in particular the PUF-AKA and PUF-MAC. In Sect. 4, we provide a security analysis and evaluate the performance. Finally, Sect. 5 concludes the paper.

## 2 System Model

### 2.1 Network Model

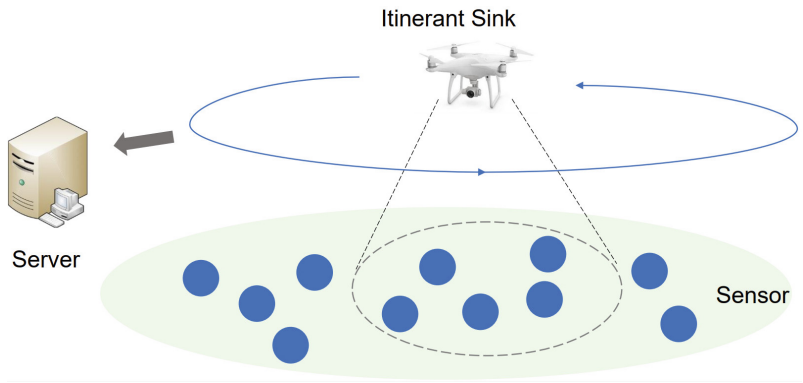



**Fig. 1.** Network model of UWSNs

The network model of UWSNs is shown in Fig. 1. This model consists of three entities: a set of sensors $\{s_1, s_2, ..., s_n\}$, an itinerant sink, and a trusted server. Salient details and assumptions are as follows:

- **Sensors** are installed in an impromptu manner with uniform distribution in the coverage area. They do not need to communicate with each other. Assuming that every sensor is equipped with a PUF that possesses enough CRPs.
- **Itinerant sink** is a mobile device (Unmanned Aerial Vehicle, Autonomous Underwater Vehicle, soldiers, etc.) to collect sensed data from sensors. It will patrol periodically over a certain geographical area and visit the available sensors. Compared with sensor nodes, sink has higher computing resources and more energy. Assuming that the sink is enclosed in a self-destructive envelope to against physical attack. We also assume it has the ability to establish a long-term secure channel with a trusted server.
- **Sever** is the only trusted authority in this network. We assume that CRPs of all sensors have been registered on the server in network initialization phase. This process can be done using physical layer key extraction scheme [16] or time-based one-time password algorithm (TOTP) [17].
- **Data collection:** in the data collection task, the unit of time is rounds. Each sensor synchronously collects a single data per round. Itinerant sink periodically visits the sensors to collect data at most $v$ rounds apart. Assuming that the sensors have enough storage to save these $v$ rounds data.
- **PUF** is implemented on the off-the-shelf SRAM so that it does not cause extra hardware costs. We assume that PUF in each sensor is unique and the communication between each sensor's MCU and its PUF is secure.

### 2.2 Adversarial Model

We assume that the ADV in our scheme focuses on modifying or manufacturing data in UWSNs rather than deleting or delaying data. This assumption is classical, which can referred to [2,9]. The ultimate goal of ADV is to introduce a single fraudulent data in device $s_i$ at round $\overline{r}$. The security mechanism has no idea about $s_i$ and $\overline{r}$, so we need to be concerned with all the data collected at all rounds. ADV will achieve success if the data he injected finally becomes the measurement of $s_i$ at round $\overline{r}$ through the data authentication scheme. Otherwise, the ADV fails. Emphasizing that we do not consider the problems of data survival [3], which is another branch of UWSNs data security.

Our adversary model has the following capabilities:

- **Compromising capabilities**: ADV can compromise any number of sensors at any round, because our scheme does not rely on the co-operation mechanism and sensors protect their data by themselves. ADV can acquire any secret information stored in memory and monitor all the communications incoming and outgoing of the compromised sensor.
- **Proactive**: ADV starts to compromise sensors at round 1, before receiving any information about the target sensor and the target data collection round.
- **Minimal Disruption**: ADV does not interfere with sensor behavior. So it will be undetected.
- **Network knowledge**: ADV knows the composition and the topology of the network.

# 3 Proposed Practical Data Autnentication Scheme

## 3.1 PUF-AKA Protocol

**Initialization:** Session key $S_{k1}$ has been established between itinerant sink and trusted server. Each sensor has registered its round 0 information (including $ID_{Init}$ and $(C_0, K_0)$) on the server. Server maintains a $RigestList$ with $n$ key-value pairs $(ID, (C, K))$. A hash function $Hash$ and two non-linear functions $F_1$, $F_2$ have been implemented which is public to everyone (Fig. 2).

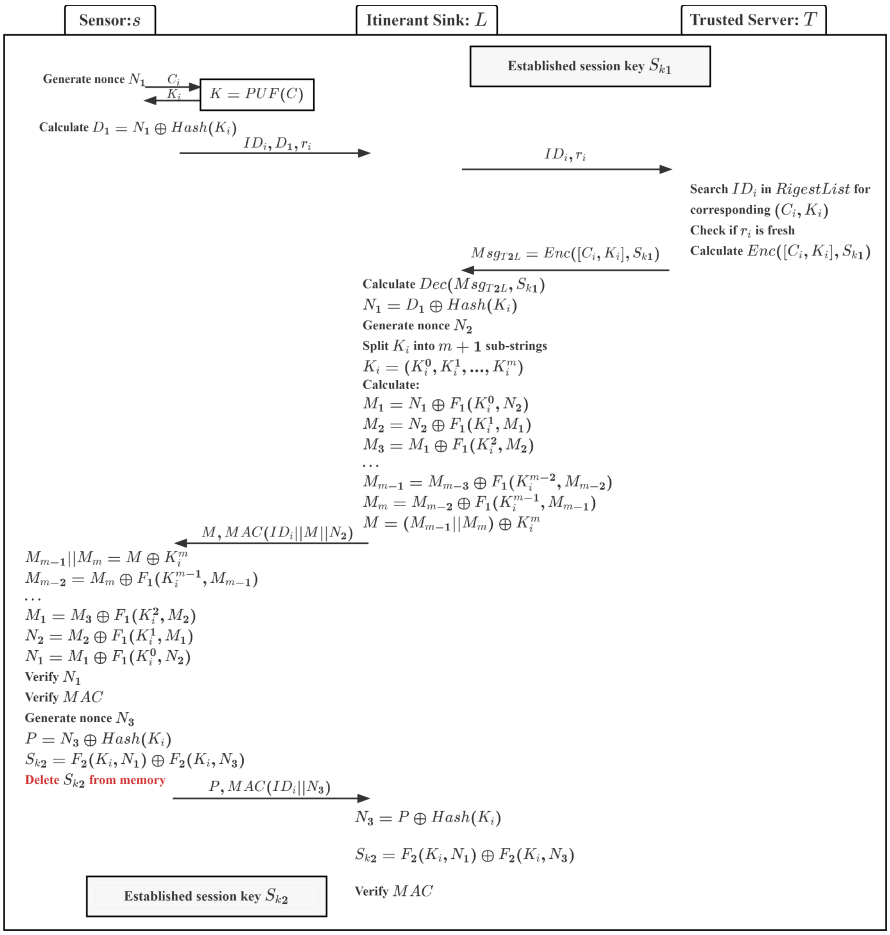



**Fig. 2.** Authentication and key agreement protocol between sensors and the sink

**Authentication and Key Agreement Protocol:** Suppose that the sensor $s$ visits the sink for $i$ times before round $r_i$. If $s$ wants to visit sink in round $r_i$, it

generates a nonce $N_1$ and obtains a CRP $(C_i, K_i)$ through PUF. Then it sends a message to sink with $Msg_{S2L} = \{ID_i, D_1, r_i\}$, in which:

$$D_1 = N_1 \oplus Hash(K_i) \tag{1}$$

Sink $L$ needs to request the corresponding CRP from the trusted server $T$. So it forwards the message to server with $Msg_{L2T} = \{ID_i, r_i\}$. Server $T$ searches the $ID_i$ in $RigestList$ for the $(C_i, K_i)$ and encrypts the target CRP with $S_{k1}$. Then it replies a message to $L$ with $Msg_{T2L} = Enc([C_i, K_i], S_{k1})$.

When the $L$ receive the $Msg_{T2L}$, it decrypts the message with $S_{k1}$ to obtain $(C_i, K_i)$, in which $K_i$ can be split into $m + 1$ sub-strings:

$$K_i = (K_i^0, K_i^1, ..., K_i^m) \tag{2}$$

According to Eq. 1, $L$ can calculate $N_1$ as below:

$$N_1 = D_1 \oplus Hash(K_i) \tag{3}$$

Then $L$ generates a nonce $N_2$ to challenge the sensor. To encrypt the message, sink needs a simple encryption mechanism. We choose a XOR-based block encryption algorithm refer to [17]:

$$\begin{gathered} M_1 = N_1 \oplus F_1(K_i^0, N_2) \\ M_2 = N_2 \oplus F_1(K_i^1, M_1) \\ M_3 = M_1 \oplus F_1(K_i^2, M_2) \\ ... \\ M_{m-1} = M_{m-3} \oplus F_1(K_i^{m-2}, M_{m-2}) \\ M_m = M_{m-2} \oplus F_1(K_i^{m-1}, M_{m-1}) \\ M = (M_{m-1}||M_m) \oplus K_i^m \end{gathered} \tag{4}$$

After calculation, $L$ sends ciphertext $M$ along with a message authentication code $MAC(ID_i||M||N_2)$, which is used to ensure several security features. $ID_i$ is used to verify the correct sensor. $M$ is the tamper proof of message. The verification of decryption results is realized by $N_2$.

On receiving the message from $L$, sensor $s$ requests $K_i$ from PUF using challenge $C_i$. Then it decrypts $M$ with $K_i$ as shown below:

$$\begin{gathered} M_{m-1}||M_m = M \oplus K_i^m \\ M_{m-2} = M_m \oplus F_1(K_i^{m-1}, M_{m-1}) \\ ... \\ M_1 = M_3 \oplus F_1(K_i^2, M_2) \\ N_2 = M_2 \oplus F_1(K_i^1, M_1) \\ N_1 = M_1 \oplus F_1(k_i^0, N_2) \end{gathered} \tag{5}$$

$s$ verifies the validity of the message. First, it judges whether sink has passed the random number challenge through $N_1$. Then it verifies the $MAC$ to ensure the above several security traits. If it fails, $s$ will terminate the authentication. If not, $s$ generates a nonce $N_3$ and encrypts it as follow:

$$P = N_3 \oplus Hash(K_i) \tag{6}$$

Then $s$ can calculate the session key when it needs to encrypt sensitive data:

$$S_{k2} = F_2(K_i, N_1) \oplus F_2(K_i, N_3) \tag{7}$$

Emphasizing that $S_{k2}$ will only be calculated when uploading data. Once the encryption task is completed, it will be deleted from memory.

Sensor $s$ sends $P$ along with $MAC(ID_i||N_s)$ to the sink. Sink $L$ restore the $N_3$ using $P$ and $K_i$, and calculates session key $S_{k2}$ using $N_1$, $N_3$ and $K_i$. Then it verifies the $MAC$. So far, we have successfully establish a pair of session keys between sensor $s$ and sink $L$.

$$N_3 = P \oplus Hash(K_i) \tag{8}$$

$$S_{k2} = F_2(K_i, N_1) \oplus F_2(K_i, N_3) \tag{9}$$

### 3.2 PUF-MAC Scheme

After AKA process, a pair of symmetric key $S_{k2}$ has been established between sink and sensor. Sink has got the CRPs corresponding to sensors. Then we design a multi-round MAC based on the symmetric key and CRPs to ensure the data integrity.

Suppose a sensor $s$ has a connection with sink $L$ in round $t$ and the next connection will happen in round $t + v$. Sensor $s$ collects data $D_i$ in each round $i$ ($t < i \leq t + v$). Once the data is collected, MAC needs to be generated. Otherwise, ADV can tamper with the data in the memory without being found. As an example, Fig. 3 shows the specific calculation process at round $i$ and round $i + 1$. The basic framework of PUF-MAC is Encrypt-and-MAC (E&M). Considering the large bits of MAC generated by multiple rounds of data, we use a MAC chain to reduce the amount of MAC value and finally reduce the communication complexity. At round $i$, sensor $s$ generates the tamper proof messages $C_i$ by:

$$E_i = E(D_i, S_{k2}) \tag{10}$$

$$MAC_i = HMAC(D_i||MAC_{i-1}, K_i) \tag{11}$$

$$C_i = E_i||MAC_i \tag{12}$$

The symmetric encryption algorithm is AES with 256-bit key. The HMAC function is HMAC SHA-256. $K_i$ is a response generated by a PUF using $C_i$. When the scheme runs to round $t + v$, sensor $s$ finally send $R$ to sink including all the encrypted data and the MAC value of the last round.

$$R = E_t \;||\; E_{t+1} \;||\; ... \;||\; E_{t+v-1} \;||\; C_{t+v} \tag{13}$$

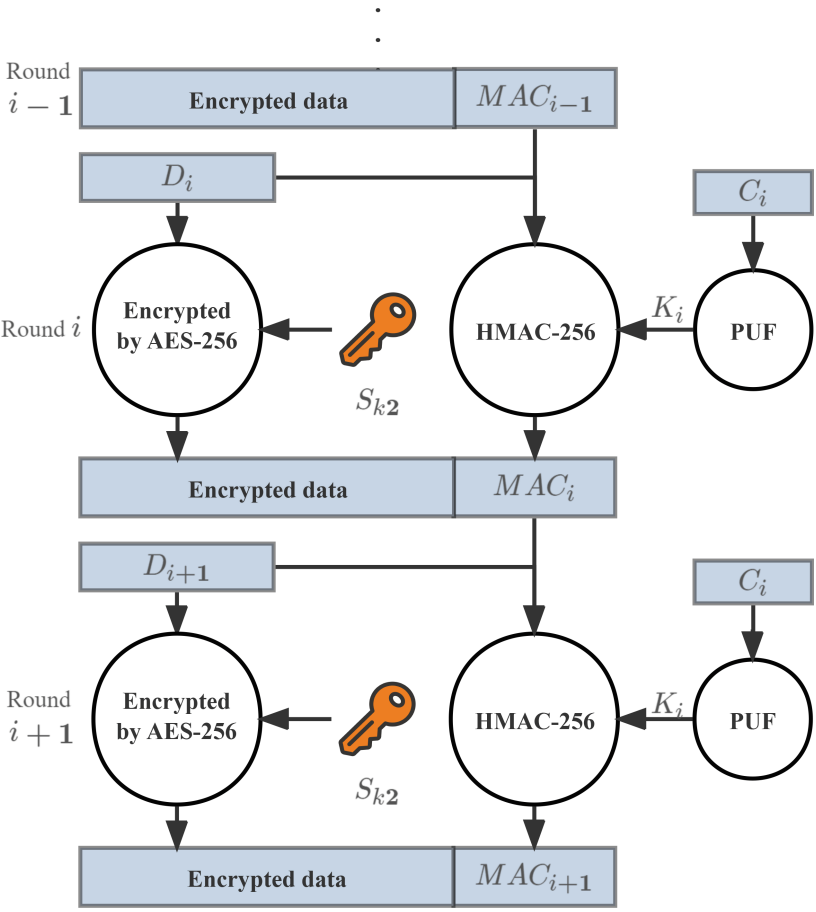



**Fig. 3.** PUF-MAC scheme

## 4 Analysis and Experiment

### 4.1 Security Analysis

The comparison of security features between our scheme and other existing works is shown in Table 1. "✓" indicates that scheme can against certain attacks or has corresponding security feature. A blank indicates this work lacks the feature.

[6,18] introduce a constantly present sink, so they are difficult to implement in UWSNs with an itinerant sink. Our scheme has a weak assumption that sink only needs to visit the sensors at most $v$ rounds apart.

Several research relies on multi-hop routing network [7,9] which will cause a relatively high communication complexity. Our scheme only establishes a direct

**Table 1.** Comparison of security feature

| Features | [6] | [18] | [7] | [9] | [4] | Our scheme |
|---|---|---|---|---|---|---|
| Ltinerant sink | | | ✓ | ✓ | ✓ | ✓ |
| Single-hop route | ✓ | ✓ | | | ✓ | ✓ |
| Proactive adversaries | ✓ | ✓ | | ✓ | | ✓ |
| Forward security | ✓ | ✓ | | ✓ | ✓ | ✓ |
| Backward security | ✓ | ✓ | | ✓ | | ✓ |
| Physical security | | | $*_1$ | $*_2$ | | ✓ |

$*_1$: The scheme has the probability to against physical attacks (the probability is related to the number of routing hops).
$*_2$: ADV can physically compromise a limited constant number $k$.

channel between sink and sensors. So we can work without complexity multi-hop route.

Compared with the collaborative solution, the security of our scheme does not rely on the network topology and the threshold of redundant storage, but only relies on the security of standard symmetric cryptographic primitives (HMAC, AES, etc.) and the robustness of PUF. This feature provides great contribution for the security of our scheme. First, due to the single-hop connection between sink and sensors, a proactive adversary can not get more information than a normal adversary. Because in our network, all sensors are completely equal without topological difference. Furthermore, the leakage of a single $S_{k2}$ will not affect the future or past data security. Because of the unclonable nature of PUF, ADV has no ability to know $(C_i, K_i)$. Sensors can periodically conduct a key update protocol based on CRPs to ensure the security of symmetric keys. So our scheme can easily achieve forward security and backward security.

The most important advantage of our scheme is unconditionally against physical attacks. Collaborative data authentication can partly meet the physical security feature. For example, [7] has the probability against physical attacks and the probability is related to the number of routing hops. [9] can ensure security when $k$ nodes are physically corrupted in each round. Our scheme assumes the communication between sensor's microcontroller and its PUF is secure because they are both on the same chip. Thus, even though ADV capture the sensors, they can not get any secrets.

### 4.2 PUF Experiment

One of the challenges of data authentication schemes for UWSNs is the implementation on commercial microcontrollers. We use off-the-shelf MCU and SRAM to obtain a set of PUFs and test its suitability as CPRs for PUF-AKA and PUF-MAC in Sect. 3.

We have implemented the PUF on two testbeds with STM32F103RBT6 microcontrollers. Each of them is equipped with a 256 kbit SRAM Cypress CY62256NLL. In total, we took 200 readings of SRAM from each testbed. The

environment temperature of testbeds was 18 °C. The working voltage of SRAM was 5 V ± 0.2 V.

The main purpose of PUF is to provide a device-specific response that can be easily extracted again. We use fractional Hamming Distance (fHD) to evaluate the percentage of different bits between PUF responses. Suppose $r_f^i$ represent the response of a PUF $f$ to the challenge $i$. $l$ is the length of response vector in bits. The fHD can be defined as:

$$fHD(r_f^i, r_g^j) = \frac{1}{l} \sum_{k=0}^{l-1} r_f^i[k] \oplus r_g^j[k] \tag{14}$$

If the responses come from the same PUF ($f = g$) and driven by the same challenge ($i = j$), then Eq. 14 represents the intra-chip Hamming distance ($intra - fHD$), which indicates the reproducibility of a PUF. If $g$ and $f$ are not the same PUF, but the responses $r_f^i$ and $r_g^j$ are driven by a same challenge ($i = j$), then Eq. 14 represents the inter-chip Hamming distance ($inter - fHD$), which indicates the uniqueness of the PUFs. In ideal PUFs, $fHD_{intra}$ should be 0 and $fHD_{inter}$ should be 50%.

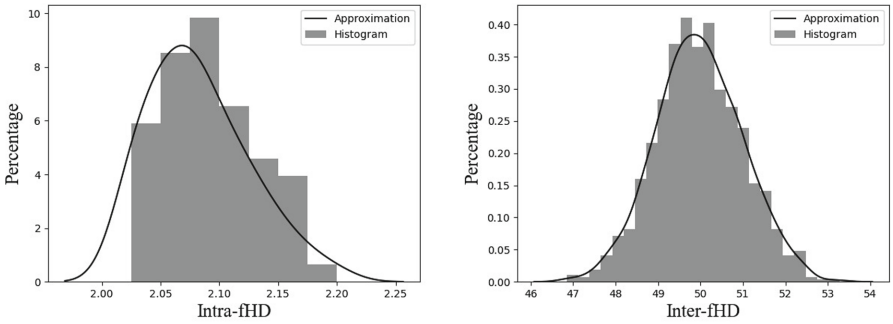



**Fig. 4.** Distribution of $intra-$ and $inter - fHD$ with a same challenge

Figure 4(a) shows the distribution of $intra - fHD$ in one chip with a specific challenge. The $intra - fHD$ can be controlled at about 2.10%, which can be easily recorrected by a fuzzy extractor ([12] has given an implementation of fuzzy extractor which can recorrect 23.8% of the data length). Figure 4(b) shows the distribution of $inter - fHD$ between two chips. The result indicates a normal distribution with an expected value 50%. It can be seen that our PUF implementation is robust and secure.

## 5 Conclusion

In this paper, we propose a practical data authentication scheme for UWSNs using PUF. Considering unattended nature of sensors, we introduce PUF to against the physical attack from adversaries. In detail, we propose a lightweight authentication and key agreement protocol based on PUF. We also propose a PUF-based MAC scheme using symmetric keys and CRPs. Compared with existing studies, our scheme has fewer network assumptions and simpler implementation. Real-world experiments show that our PUF-based data authentication scheme is effective and robust.

**Acknowledgements.** This work is supporteed by Dalian Science and Technology Innovation Fundation, NO. 2021JJ12GX013.

## References

1. Choi, H.-B., Ko, Y.-B., Lim, K.-W.: Energy-aware distribution of data fragments in unattended wireless sensor networks. In: 2018 Third International Conference on Security of Smart Cities, pp. 1–8. IEEE (2018)
2. Dimitriou, T., Sabouri, A.: Pollination: a data authentication scheme for unattended wireless sensor networks. In: 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 409–416. IEEE (2011)
3. Pietro, R.D., Mancini, L.V., Soriente, C., Spognardi, A., Tsudik, G.: Catch me (if you can): data survival in unattended sensor networks. In: 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 185–194. IEEE (2008)
4. Di Pietro, R., Mancini, L.V., Soriente, C., Spognardi, A., Tsudik, G.: Playing hide-and-seek with a focused mobile adversary in unattended wireless sensor networks. Ad Hoc Netw. **7**(8), 1463–1475 (2009)
5. Liu, D., Ning, P., Zhu, S., Jajodia, S.: Practical broadcast authentication in sensor networks. In: The Second Annual International Conference on Mobile and Ubiquitous Systems, pp. 118–129. IEEE (2005)
6. Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, J.D.: SPINS: security protocols for sensor networks. In: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom 2001), pp. 189–199. ACM (2001)
7. Ye, F., Luo, H., Lu, S., Zhang, L.: Statistical En-Route filtering of injected false data in sensor networks. In: IEEE INFOCOM 2004, pp. 2446–2457. IEEE (2004)
8. Zhu, S., Setia, S., Jajodia, S., Ning, P.: Interleaved hop-by-hop authentication against false data injection attacks in sensor networks. ACM Trans. Sen. Netw **3**(3), 1550–4859 (2007)
9. Pietro, R.D., Soriente, C., Spognardi, A., Tsudik, G.: Collaborative authentication in unattended WSNs. In: Proceedings of the 2nd ACM Conference on Wireless Network Security, pp. 237–244. ACM (2009)
10. Ravikanth, P.S.: Physical one-way functions. Ph.D. dissertation, Massachusetts Institute of Technology (2001)

11. Gassend, B., Clarke, D., van Dijk, M., Devadas, S.: Silicon physical random functions. In: The 9th ACM Conference on Computer and Communications Security (CCS 2002), pp. 148–160. ACM (2002)
12. Sajim, A.S.: Open-source software-based SRAM-PUF for secure data and key storage using off-the-shelf SRAM. Master's thesis, Delft University of Technology (2018)
13. Prada-Delgado, M.Á., Baturone, I., Dittmann, G., Jelitto, J., Kind, A.: PUF-derived IoT identities in a zero-knowledge protocol for blockchain. Internet Things **9**, 100057 (2020)
14. Chatterjee, U., et al.: Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database. IEEE Trans. Dependable Secure Comput. **16**(3), 424–437 (2019)
15. Qureshi, M.A., Munir, A.: PUF-RAKE: a PUF-based robust and lightweight authentication and key establishment protocol. IEEE Trans. Dependable Secure Comput. **19**(4), 2457–2475 (2022)
16. Zhu, X., Xu, F., Novak, E., Tan, C.C., Li, Q., Chen, G.: Using wireless link dynamics to extract a secret key in vehicular scenarios. IEEE Trans. Mob. Comput. **16**(7), 2065–2078 (2017)
17. M'Raihi, D., Machani, S., Pei, M., Rydell, J.: RFC 6238-TOTP: time-based one-time password algorithm. Internet Requests for Comments (2021)
18. Dodis, Y., Katz, J., Xu, S., Yung, M.: Key-insulated public key cryptosystems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 65–82. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_5

# On Eliminating Blocking Interference of RFID Unauthorized Reader Detection

Degang Sun[2], Yue Cui[1,2], Siye Wang[1,2(✉)], and Yanfang Zhang[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
wangsiye@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

**Abstract.** RFID as an important component technology of IoT is rapidly applied in recent years. But it also faces severe security risks like malicious intrusion, most operated by unauthorized reader (UR). There are some researches proposed the unauthorized reader detection algorithms based on commercial off-the-shelf (COTS) devices, but these detection algorithms are often easily affected by moving objects blocking interference, causing false alarms. We adopt a new parameter adjacent signals time interval (ASTI) to improve the UR detection algorithm by reducing the time-delay and propose a new method of eliminating moving object interference, which can reduce the system false alarm rate to less than 7.9% by experimental testing.

**Keywords:** RFID security · Intrusion detection · Human-centered computing · Privacy protection

## 1 Introduction

IoT has gained more and more attention in recent years for its rapid development. RFID, as one of the most important IoT technologies, is also applied in many fields like, supply chain [9], retail business [10], identification [11] and etc.

RFID devices can be divided into two categories: commercial off-the-shelf (COTS) devices based on common protocols and customized devices using special protocols. Specialized devices are usually deployed for special demand and what we use most in daily life are COTS devices. The rapid promotion of RFID can't be achieved without the support of common protocols. In UHF RFID, there is one common protocol proclaimed by the standard organization: ISO 18000-6C [1] (EPC C1 G2). The devices can communicate with each other as long as they use the same protocol.

But ISO 18000-6C protocol used by the COTS devices lacks security certification between the tags and the readers. So any unauthorized reader in the space can communicate with the tags and get access to the tags data, resulting in data leakage, data tampering, and other security threats [2]. Due to the reader is the main way to obtain the UHF RFID air interface data, it means unauthorized


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 591–602, 2022.
https://doi.org/10.1007/978-3-031-19208-1_49

reader (UR) detection is the major concern for intrusion detection of the UHF RFID system.

Most researches about RFID security rely on customized devices or arranging special protocols. But these method faces cost and scalability concern when deployed. Using the COTS reader itself as monitoring equipment, without other attachment, is a new trend of researches.

COTS reader can obtain the data like received signal strength indicator (RSSI), Phase, timestamp, throughput rate. Some UR detection researches [5,6] used throughput rate and achieved good performance. But one of the problems of throughput rate is the alarm time-delay because the UR detection system has to accumulate data within the set time window. We found a new parameter ASTI (adjacent signals time interval) to make the UR detection system more sensitive to the abnormal data. When using these methods in the office environment, we found that the surrounding pedestrian walking or blocking (like left part of Fig. 1) will affect the signal, causing a false alarm. It inspired us to eliminate the blocking interference of these moving objects.

Radio-based human sensing is a hot topic in recent years, mainly focusing on activity recognition [12] and localization [13]. However, the ID or gesture of a person or object blocking our RFID system is not important. We made a hypothesis that velocity is the key for eliminating the interference and arranged experiments to prove it.

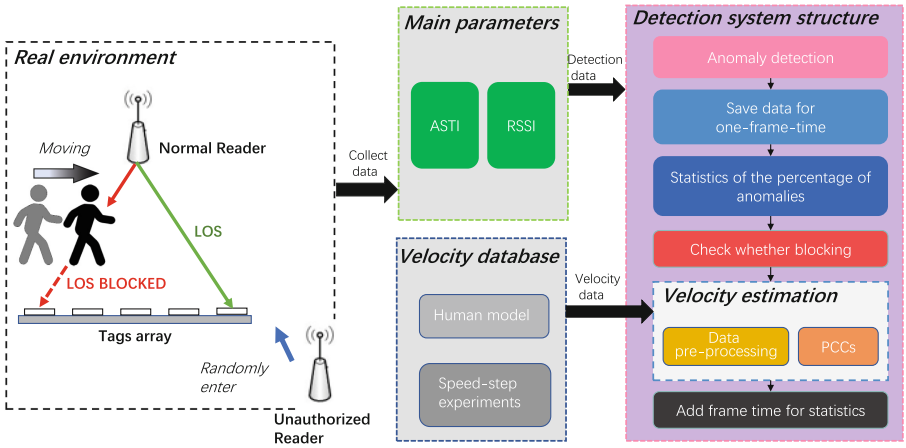



**Fig. 1.** System architecture overview

Figure 1 is the architecture of our system. In the experiments, we deployed one normal reader and tags array (five tags equally spaced at the board in front of the reader). Normal reader was set at default settings. The UR randomly entered and started. Sometimes, moving object passed by and blocked the line-of-sight (LOS) path between reader and tags. The normal collected data including our

main parameters: ASTI and RSSI. When anomaly was detected, the system began to save one-frame-time data for UR detection algorithm and found out if alarming. Next, the interference eliminating part worked. First, checking whether there was object blocking. If not, we determined the UR occurred. If object blocking existed, we used velocity estimation algorithm combined with velocity database to calculate the interference time of blocking. Then added interference time to the detection frame and data within that time to UR detection algorithm. If the system still alarmed, that's the real alarm caused by UR.

The main contributions of this work are as follows:

1. This is the first work using ASTI for unauthorized reader detection algorithm which significantly reduces the time-delay compared with the algorithm using throughput rate.
2. Analysing model of human moving through RFID system and relating the blocking time with the energy of RFID transmitting signal of multipath put up an assumption line-of-sight (LOS) is the main reason for blocking interference.
3. We propose a method based on velocity estimation for eliminating the interference to the UR detection. Extensive experiments verify our system. The results show perfect performance.

## 2 Related Work

Most RFID security researches focus on specialized protocol or data encryption such as Pramod present physical unclonable function-based unilateral authentication protocol for RFID system [3]. The algorithm has been implemented on an 8-bit open-loop resonator-based chipless RFID tag-based system and is validated using BASYS 2 FPGA board-based platform [4]. But these methods rely on customized devices or additional chips and will increase the cost of the system.

Due to the open-air interface transmitting of the UHF RFID system, the signals can be captured using Universal Software Radio Peripheral (USRP). Based on these physical layer signal characteristics, many researchers like Savry [14], Katabi [15] Ding [16] have proposed a variety of effective physical layer security mechanisms. The major drawbacks of these studies are expensive devices and scenario customization.

There are also some security researches using data from COTS RFID. Huang [5] proposed an unauthorized reader detection system based on throughput. Then Sun [6] managed to trace the unauthorized reader using deep reinforcement learning. Razm [17] introduced fuzzy logic into the RFID intrusion detection. However, they all face the challenge of time-delay.

## 3 Unauthorized Reader Detection System Analysis

In this section, we will analyze the RFID unauthorized reader (UR) detection system based on ASTI (adjacent signals time interval) which is an optimal parameter compared to throughput rate because of lower time delay [5]. In addition,

according to the needs of actual use, we chose a common scenario to arrange the experiment - placing the reader at the corridor (as Fig. 1). And when a human or other object passes by, how does the performance of the UR detection system change.

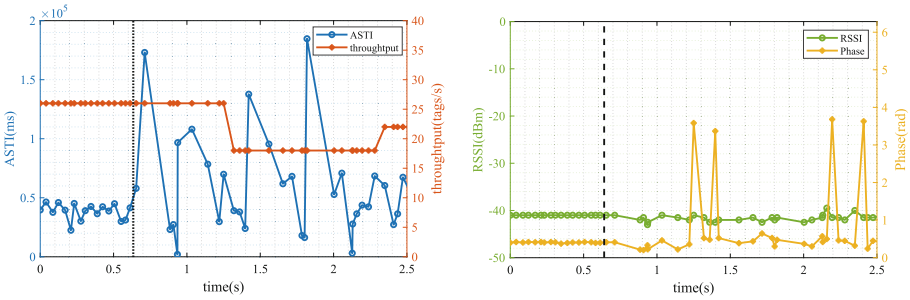
### 3.1 Analysis of the Parameter in UR Detection System

We set up the pre-experiment as Fig. 1, the normal reader was set at the corridor and started up to communicate with the five tags equally spaced at the board in front of the reader. Then the UR randomly entered and started.

The reader communicates with the tags and receives the signal reflected from tags. Once the signal is received, the reader records the time (Timestamp). Received signal strength indicator (RSSI) is a measurement often used in the RFID system of the power present in a received radio signal. Adjacent signals time interval (ASTI) means the time interval of signals received by the reader and is calculated by Eq. 1:

$$ASTI = Timestamp(i) - Timestamp(i-1) \tag{1}$$

of which, $Timestamp(0)$ is the set as 0 that means the time reader starts to work, $i \geq 1$.




(a) ASTI & throughput (b) RSSI & Phase

**Fig. 2.** Parameters under unauthorized reader

Figure 2 shows one of the results, the black dotted line indicates the time UR starts. After UR starts, the variation of ASTI and throughput are both significant but the former one at a lower time-delay. As for the RSSI and Phase, we can see from Fig. 2(b) that RSSI varies as small fluctuations and the Phase fluctuates up and down around the stable value except there are some sudden peaks. So the change rate of ASTI can be adopted as a parameter for UR detection.

In the next part, we started the normal reader and UR, then a man walked through the corridor to simulator the true environment for daily use. Figure 3

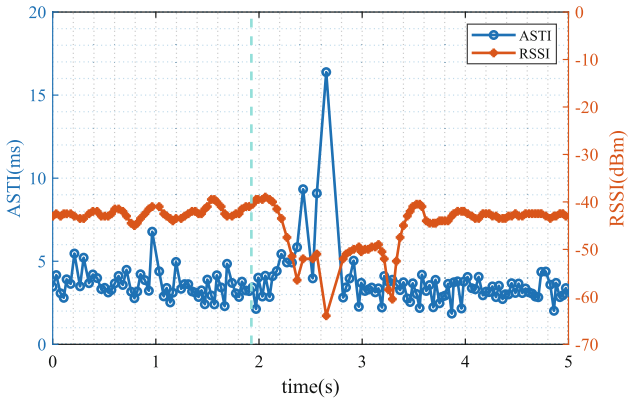



**Fig. 3.** ASTI & RSSI under unauthorized reader and passing-by

shows one of the experiment results. The light blue dotted line indicates when the man blocks the LOS path between tags and normal reader.

In general scenarios, the parameter more commonly used for human sensing is Phase. However, in cases with UR, as shown in Fig. 2, the phase signal plots will irregularly and suddenly occur a peak, which makes it inaccurate when using this indicator for sensing tasks.

From the result, we can find that ASTI also changes noticeably as humans passing so does the RSSI. But because the RSSI is not sensitive to whether UR is at present, we can use RSSI as the parameter for human-centered computing.

Human passing-by and UR cause the same result for the ASTI UR detection so the false-alarm rate (FPR) will be high in the real environment. It's vital to find a way to eliminate the interference of humans.

### 3.2 UR Detection System

This is the key part of the RFID intrusion detection system. Huang et al. [5] proposed a UR detection system based on the change rate of throughput. But as the previous experiments have shown, the system based on throughput suffers from the time-delay. Taking advantage of the new parameter ASTI, we propose a new algorithm for the RFID UR detection system.

**Anomaly Detection:** Our goal is to detect the UR while the normal RFID system working. First, we collect the data in a steady environment named $ASTI_S$ as the reference, set a threshold $\theta_R$ according to the change rate $R$ from $ASTI_S$ when UR appears. The normal reader received data including ASTI, if the change rate $R$ is larger than $\theta_R$, there is an anomaly value.

$$R = \frac{ASTI - ASTI_S}{ASTI_S} \tag{2}$$

**Save Data for One-Frame-Time:** When a anomaly value occurs, the UR detection system begins to collect and save data into an array $Risk[]$, which data

pattern is like $Risk[i] = [Timestamp[i], ASTI[i], RSSI[i]]$. The $Timestamp$ of the first data is $Timestamp[0]$, which is the sign of $frame$ beginning. We define a time window named $frame$ to indicate the system save how much data for the detection.

**Statistics of the Percentage of Anomalies:** We think the system has obtained enough data for the detection and the storage is stopped, if $Timestamp[i] - Timestamp[0] \geq frame$. Then the system calculates how many data points in $Risk[i]$ called $N$ and in which how many data points are anomaly value by Eq. 2 called $A$. Define a threshold $\theta_a$, if the rate of $\frac{A}{R} \geq \theta_A$, there appears an unauthorized reader, otherwise ignores this alarm.

Some indicators need to be set before the RFID UR detection system starts to work: the steady value in environment $ASTI_S$ two thresholds $\theta_R$ and $\theta_A$. All of them can be calculated trial and error by pre-experiments.

### 3.3 Model of Moving Human

First, we model and analyze the action of a moving object as an example with top view like Fig. 4. The reader antenna is deployed on the left and the tags array (3 tags in the picture) is deployed on the right side. The blue square is treated as a pedestrian with size of length $l_h$ and width $d_h$ respectively. He passes the corridor in the middle at speed $V$m/s. The corridor distance from left to right is $d_{lr}$. The tags are equally spaced at $d_{set}$, the effective length of the signal reception of the tags is $l_{tagi}$, the red line part is the angle of the human body completely blocking the LOS path between tag1 and the antenna, the corresponding impact distance of $d_{in}$ and impact time $T_{in}$ can be calculated as follows:

$$d_{in} = \frac{\frac{1}{2}(d_{lr} + l_h)}{tan(\frac{\pi}{2} - \theta_{tag1B})} - \frac{\frac{1}{2}(d_{lr} - l_h)}{tan(\frac{\pi}{2} - \theta_{tag1A})} + d_h \tag{3}$$

$$T_{in} = \frac{d_{in}}{V} \tag{4}$$

In the RFID system, the signal received by the reader antenna can be divided into line-of-sight (LOS) signal and multipath effect signal superposition caused by other reflection paths.

$$H(f) = \sum_{n=0}^{N-1} \rho_n e^{j\theta_n} e^{-j2\pi f \tau_n} \tag{5}$$

Equation (5) represents the superposition of the arrival signals of N paths. Where: $\rho_n$ denotes the intensity of the signal; $\theta_n$ denotes the phase of 'n' path; $\tau_n$ denotes the arrival delay. In general, the signal energy of the line-of-sight path (the straight-line path between the sender and the receiver, denoted as n = 0 part). $\rho_n$ in Eq. 5 is much larger than the signal energy of the other paths and thus dominates in the superposition dominant position. In an ideal environment, when multipath effects do not exist, it can be considered that $T_{in} \doteq \Delta t_{tagi}$.

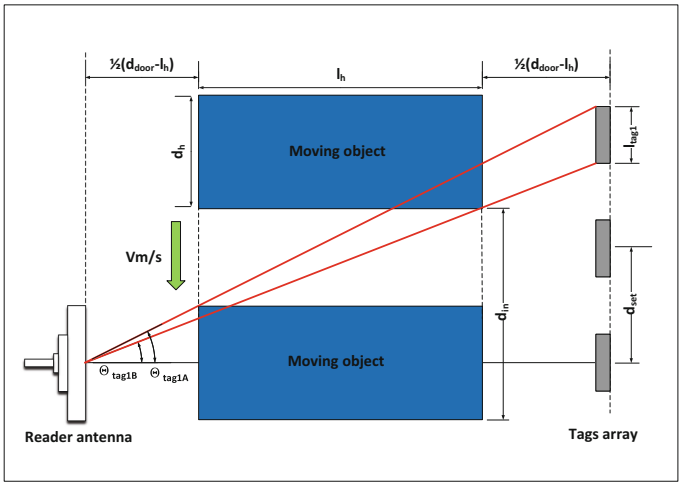



**Fig. 4.** Model of moving object passing RFID system

In practice, the tag acquires multivariate data due to environmental multipath and the passage of objects generating new multipath effects. Before the object blocks the LOS path, new multipath will be generated with the object moving until the LOS path is blocked which means the most energy tags received are lost. Then the object moves forward and it's a mirroring process, so the signal changes will be mirroring similar to the former.

## 4 Eliminating Blocking Interference Method

From the previous experiments and modeling analysis, we figure out the interference time of passing by depends on the velocity of the object and RSSI is the best metric for velocity sensing. So in this section, we will introduce the method using RSSI for velocity estimation and eliminating the blocking interference.

### 4.1 Data Pre-processing

ALL RSSI data received by reader should be pro-processing first. We use a Moving Average Filter (MAF) to smooth the acquired data for further processing. And due to its polling mechanism, the RFID reader can only communicate with one tag at a time and the sampling frequency of commercial readers may only 30 Hz [18] in the environment, however, since the object is always in motion, we want to capture more readings at the same time. Therefore, we use the Hermite Interpolating Polynomial (PCHIP) method to interpolate the data samples at the desired point in time, which allows for more accurate interpolation at a considerably higher efficiency.

### 4.2 Velocity Database

In the beginning, a database of moving object speed should be built as the reference for matching. To achieve this, we arranged a dummy placing at a vehicle fixed on the slide rail, which would move in the set direction with a set velocity. We used the machine to simulate real humans or other objects passing through the door like Fig. 1. The velocity is raised in steps of 0.02 from 0.2 m/s to 0.8 m/s (interval of daily human walking speed [8]) and at each velocity tested for 200 times than taking the average number as the result.

We use Pearson correlation coefficient [7] for velocity matching. The Pearson correlation coefficient between two variables is defined as the product of the covariance of the two variables divided by their standard deviations:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{6}$$

Because the calculation of PCCs only needs $(Y - \mu_Y)$ and $\sigma_Y$ which means that the setup of velocity database needs a two-dimensional array for each velocity.

### 4.3 Eliminating Blocking Interference

When the UR detection algorithm sends an alarm, the system has to decide whether there is someone passing through causing false alarm or real UR existing. The velocity estimation algorithm will use the RSSI data saved in one-frame-time.

From the pre-experiments, we found out that the change rate of RSSI with UR is less than 5%. So it can be used as a threshold to judge object passing-by.

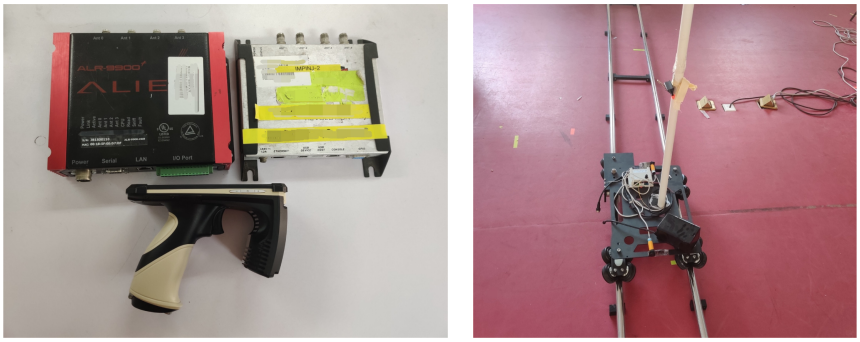
Then, the RSSI data will be pre-processed and use Pearson correlation coefficient for matching its velocity to the database. We select the velocity which PCC is the closest to 1 as the estimation of the velocity of the moving human or object. And $T_{in}$ can be calculated by Eq. 4.

In summary, the moment the UR detection system sends an alarm, the RSSI change rate is calculated and if the rate is larger than 5%, the velocity estimation part work to match the object speed and get $T_{in}$.

At last, $T_{in}$ the time of UR detection influenced add the set window time will be thought as the new window time back to UR detection system: $frame = frame + T_{in}$. If the system still alarms, We think UR is shown up.

## 5 Experiment

In this section, we arranged some experiments to evaluate our method. The experiment environment was a normal office, devices setup like Fig. 1, tested 500 times each group. We choose several devices that are often applied in the commercial scene: two Impinj R420, one alien 9900 and one handle reader ORCA-50 as readers (one Impinj R420 for the normal reader and others for unauthorized readers) and tags are of H47. One vehicle with a railway and a dummy to move at a set speed (Fig. 5).



(a) readers (b) vehicle on the railway

**Fig. 5.** Experiment devices

### 5.1 Accuracy of UR Detection System

We tested the accuracy of the UR detection system using three kinds of unauthorized readers in the three conditions:

- without human passing-by: free
- with human passing-by: interference
- with human passing-by and eliminating method: eliminated

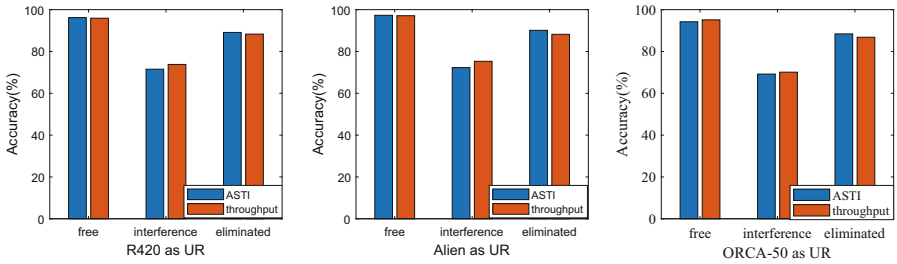



**Fig. 6.** Accuracy of UR detection system

The result shows in Fig. 6. For accuracy, the system with ASTI and throughput performs very close, the accuracy for detecting different kinds of unauthorized readers shows only minimal changes. When facing interference, the system with throughput act better. And our method can reduce the interference of human passing-by for both systems obviously.

### 5.2 Time-Delay of UR Detection System

In this section, we tested the time-delay of the UR detection system with ASTI and throughput. We set four different length frames for experiments divided

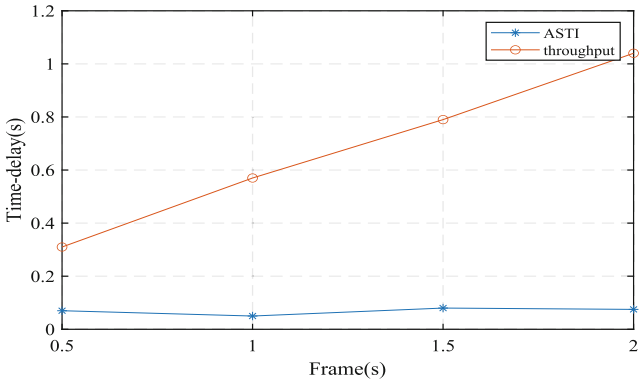



**Fig. 7.** Time-delay of two system based different parameters

into four groups. From the previous experiments it can be seen that different UR doesn't affect the performance of the system, so we just use R420 as UR. Each group was intruded on by UR 100 times. Results are depicted in Fig. 7. With a longer frame, the time-delay of the throughput system also gets longer but the ASTI system keeps at 0.08s nearby.

### 5.3 Evaluation of FPR

Because the moving human affects the FPR mostly, so in this experiment, we tested for evaluation of FPR which is the most concern of our method. It's can be seen from Fig. 8, our eliminating method significantly lower the FPR compared to the unused one at 7.9%.

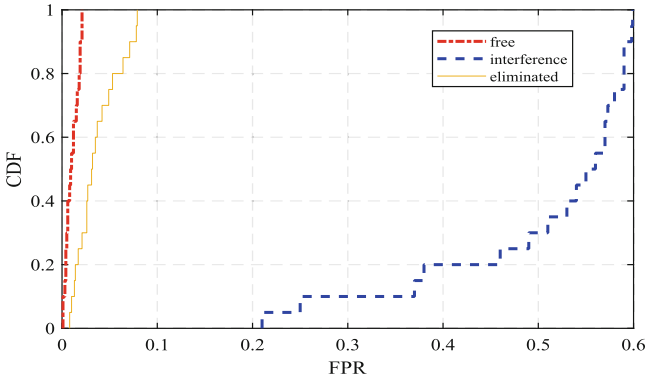



**Fig. 8.** Evaluation of FPR in three conditions

### 5.4 Accuracy of Velocity Estimation

In this experiment, we arranged an experiment for testing the velocity estimation part. Because the resolution of the velocity database is only 30, we define the accuracy of velocity estimation that if the velocity predicted value of the estimation system is the closest to any other values in the velocity database, we think it's a correct estimation. We arranged four groups of experiments 150 times for each group. High accuracy has occupied in all four groups of the experiments shown in Fig. 9.

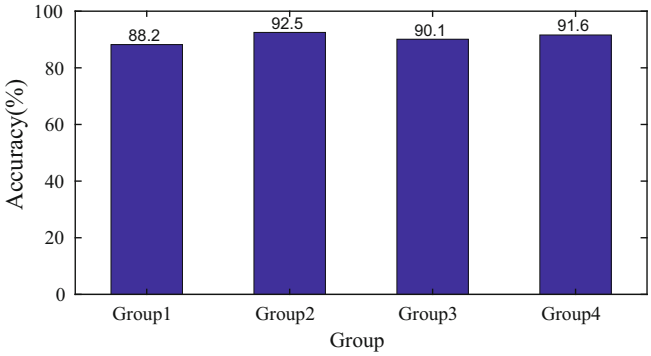



**Fig. 9.** Accuracy of velocity estimation

## 6 Conclusion

In this paper, we propose a new parameter and a method of eliminating the human or other object moving in the area causing false alarm for the unauthorized reader detection system. At first, analyze the parameters obtained in COTS RFID readers and select ASTI for the UR detection system and modeling moving humans, then find the key of velocity estimation. Next, by building a speed database and using PCCs, put forward the method to eliminate the interference. At last, test our method through experiments and the results show excellent performance. In the future, we will deploy our system in different environment to test the transfer scenario capability and explore new velocity estimation method.

**Acknowledgments.** This paper is supported by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDC02040300.

## References

1. Iso [EB/OL]. https://www.iso.org/home.html
2. Lawson, N.: Side-channel attacks on cryptographic software. IEEE Secur. Priv. **7**(6), 65–68 (2009)

3. Maurya, P.K., Bagchi, S.: A secure PUF-based unilateral authentication scheme for RFID system. Wireless Pers. Commun. **103**(2), 1699–1712 (2018). https://doi.org/10.1007/s11277-018-5875-2
4. Sharma, V., Vithalkar, A., Hashmi, M.S.: Lightweight security protocol for chipless RFID in internet of things (IoT) applications. In: 2018 10th International Conference on Communication Systems & Networks (COMSNETS) (2018)
5. Weiqing, H., Chang, D., Yue, C., et al.: RFID air port intrusion detection technology based on malicious reader discovery. Acta Sinica Sinica, 7 (2018)
6. Sun, D., Cui, Y., Feng, Y., Xie, J., Wang, S., Zhang, Y.: URTracker: unauthorized reader detection and localization using COTS RFID. In: Liu, Z., Wu, F., Das, S.K. (eds.) WASA 2021. LNCS, vol. 12937, pp. 339–350. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85928-2_27
7. Zeinali, M., Shahmorad, S., Mirnia, K.: Hermite and piecewise cubic Hermite interpolation of fuzzy data. J. Intell. Fuzzy Syst. **26**(6), 2889–2898 (2014)
8. Samra, H.A., Specker, B.: Walking age does not explain term versus preterm difference in bone geometry. J. Pediatr. **151**(1), 61–66.e2 (2007). ISSN 0022-3476. https://doi.org/10.1016/j.jpeds.2007.02.033
9. Angeles, R.: RFID technologies: supply-chain applications and implementation issues. Inf. Syst. Manag. **22**(1), 51–65 (2005)
10. Roussos, G.: Enabling RFID in retail. Computer **39**(3), 25–30 (2006)
11. Zhang, Q., Zhao, R., Li, D., et al.: Unobtrusive and robust human identification using COTS RFID. Comput. Netw. **166**, 106818 (2020)
12. Zhao, M., Li, T., Abu, Alsheikh, M., et al.: Through-wall human pose estimation using radio signals. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7356–7365 (2018)
13. Khan, U.M., Venkatnarayan, R.H., Shahzad, M.: RFMap: generating indoor maps using RF signals. In: 2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), pp. 133–144. IEEE (2020)
14. Savry, O., Pebay-Peyroula, F., Dehmas, F., Robert, G., Reverdy, J.: RFID noisy reader how to prevent from eavesdropping on the communication? In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 334–345. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74735-2_23
15. Hassanieh, H., Wang, J., Katabi, D., et al.: Securing RFIDs by randomizing the modulation and channel. In: 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15), pp. 235–249 (2015)
16. Ding, H., Han, J., Zhang, Y., et al.: Preventing unauthorized access on passive tags. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 1115–1123. IEEE (2018)
17. Razm, A., Alavi, S.E.: An intrusion detection approach using fuzzy logic for RFID system. In: Advances in Information Science and Applications, p. 2 (2014)
18. Chen, Z., Yang, P., Huang, G., et al.: RFdesk: record your objects on desktop using COTS RFID devices contactlessly. In: 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), pp. 556–561. IEEE (2019)

# Gradient-Based Adversarial Attacks Against Malware Detection by Instruction Replacement

Jiapeng Zhao[1,2], Zhongjin Liu[3], Xiaoling Zhang[1,2], Jintao Huang[1,2], Zhiqiang Shi[1,2(✉)], Shichao Lv[1,2], Hong Li[1,2], and Limin Sun[1,2]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{zhaojiapeng5035,shizhiqiang}@iie.ac.cn

[3] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, China

**Abstract.** Deep learning plays a vital role in malware detection. The Malconv is a well-known deep learning-based open source malware detection framework and is trained on raw bytes for malware binary detection. Researchers propose adversarial example generation strategies to evade the Malconv by modifying the PE headers or the end of malware. However, these strategies that focus on non-executable portions can be easily pre-processed before classification. Therefore, we propose a new instructions replacement strategy to overcome these flaws. This paper reviews the research progress on adversarial example generation strategies for the Malconv in recent years, analyzes the reason why the Malconv can be evaded by adversarial examples and identifies two layers of the Malconv that can be attacked, and propose the gradient-based instructions replacement strategy named EFGSM that is an enhanced Fast Gradient Sign Method (FGSM), and sheds light on future work in adversarial example defense strategies for the Malconv. The paper assesses the performance of our EFGSM and existing adversarial example generation strategies upon 200 malware. The results of the evaluation show that our strategy improves the success rate from 68% to 81.5% and takes less time to generate malware examples. The paper also assesses the evasion performance of adversarial examples in three antiviruses. The results depict that our strategy is the state of the art.

**Keywords:** Adversarial example · Gradient attack · Align instruction · Malconv


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 603–612, 2022.
https://doi.org/10.1007/978-3-031-19208-1_50

## 1 Introduction

Machine learning and deep learning have been widely used in the field of malware detection and have already achieved remarkable improvements. However, there is still some specialized and custom-built malware that can evade deep learning detection. Researching adversarial malware examples can discover the defects of the deep learning detection model, and then be used to adjust its architecture to avoid being evaded by hackers. Meanwhile, training with a large number of adversarial examples can improve the detection strength and robustness of the deep learning detection model.

Due to the constraints of the PE file structure, it is extremely restrained to modify malware samples. Related works have evaded static malware detection models by adding or modifying special areas that do not affect program execution. Many strategies [1–5] have been made in the research of adversarial examples for the Malconv models. According to the different areas, their strategies can be divided into PE headers, slack regions between sections, and the end of PEs. These strategies focus on dealing with bytes at non-executable portions of the PE file. Thus, the adversarial examples can be pre-processed before classification, by removing NOPs and irrelevant information.

In the paper, we propose a new strategy named EFGSM which is an enhanced Fast Gradient Sign Method (FGSM) to modify the align instruction in the text section to overcome the flaw. Through FGSM [6], we can avoid the high time overhead caused by multiple iterations. This paper assesses the performance of our EFGSM and existing adversarial example generation strategies in 200 malware. The results of the evaluation show that our strategy improves the success rate from 68% to 81.5% and takes less time to generate malware examples. The paper also assesses the evasion performance of adversarial examples in three antiviruses. The results depict that our strategy generates a higher percentage of samples that can evade the antivirus than existing strategies. The main contributions of this paper are summarized as follows:

- The paper analyze the reason why the Malconv can be evaded by adversarial examples and use the EFGSM to modify the align instructions to evade the Malconv.
- Our experiments demonstrate that our strategy can generate adversarial examples in less time and an evasion rate of 81.5%, while also being effective on deep learning-based antivirus.
- According to our strategy, we propose defenses to overcome the shortcomings of Malconv.

## 2 Background

### 2.1 Malconv for Malware Detection

This section mainly introduces the framework of the Malconv [7], a byte-based convolutional neural network detection model. The architecture diagram of the

Malconv is shown in Fig. 1. The input file size is bounded to 2 MB. If its size is smaller than 2 MB, the file is padded with the value 0xff. If it is bigger than 2 MB, only the first 2 MB are analyzed. The first layer of the network is an embedding layer, it maps byte to an 8-dimensional embedding matrix. Then the embedding matrix is divided into two matrices. One of the two matrices enters the convolution layer and the other enters the convolution layer with sigmoid. The results are multiplied and passed through a temporal max-pooling layer and a final fully-connected layer. The output $F(x)$ is given by a softmax function.

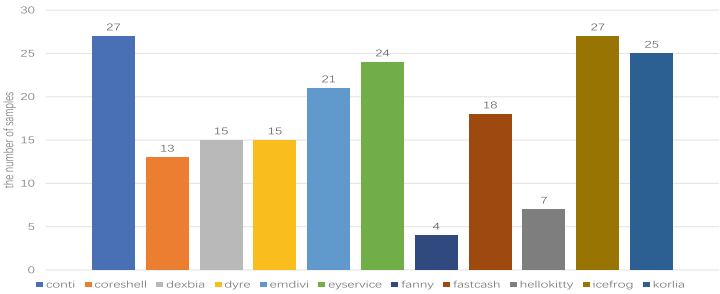



**Fig. 1.** The framework of the Malconv [7]

## 2.2 Adversarial Malware Example

Adversarial examples, proposed by CHristian Szegedy et al. [8], are specialized inputs created to get the misclassification of a given input. The adversarial malware example includes padding data into the PE file or changing the PE file data to generate a new example, causing misclassification (Fig. 2).

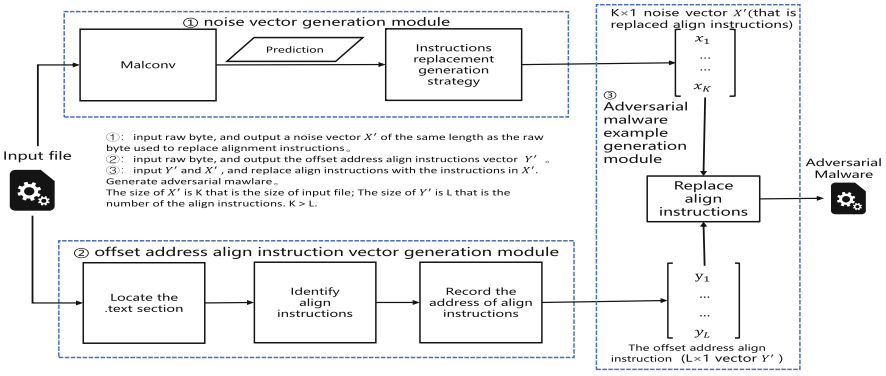



**Fig. 2.** The framework of our work

## 3 Methods and Techniques

The details of our strategy are described in this section. The summary of our strategy is shown in Fig. 2. Our strategy is divided into three parts. The first part is the foundation of our approach. It takes the raw bytes and creates the noise vector $X'$, which is used to replace instructions. The second part takes the raw bytes and generates an offset address align instruction vector $Y'$ that contains all of the align instructions' offset addresses. The final part generates adversarial malware using the vectors $X'$ and $Y'$. In the first part, we take the raw bytes to get the prediction from the Malconv. According to the prediction, we use the instruction replacement strategy to create a noise vector. Using different instructions replacement generation strategies will affect the efficiency of the final sample, so this is the core of our strategy.

In Sect. 3.1, We introduce two instruction replacement generation strategies that replace align instructions with benign sequences, coming at a considerable penalty in terms of time. In Sect. 3.2, we introduce the EFGSM that can get the best results.

### 3.1 Benign Sequences Instructions Replacement

We try a similar strategy to Luca et al. [2] which inserts benign instructions into parts of the areas that are not executed at run time. Replacing with benign instructions into every align areas may results in different prediction values. Thus, our strategy can then be equivalent to solving the globally optimal solution from the locally optimal solution. We use the number of iterations as a baseline to assess feasibility, with each iteration taking 0.1 s to get the prediction value for malware (filesize = 200 KB). It is almost impossible to directly solve the globally optimal solution because it need cost m $*$ n iterations (where m is the number of align areas, n is the number of replaced benign instructions sequences). We consider using "continuous replacement" and "repeat replacement". We choose 100 benign files and extract 300 benign instruction sequences from functions whose prediction values are less than 0.5.

***Repeat Replacement***. We use "repeat replacement" strategy that replaces all align instructions with the same benign instructions sequences. Taking the malware [1] as an example, the malware has 34 align areas and it costs 300 iterations to get the benign instructions sequences that reduce the maximum prediction value. Although the entire replacement takes 30 s to complete, it only decreases the prediction value from 0.97 to 0.6 in Fig. 3.

[1] MD5:c37b02e060fa169e4d6f0c6e77ddb500.

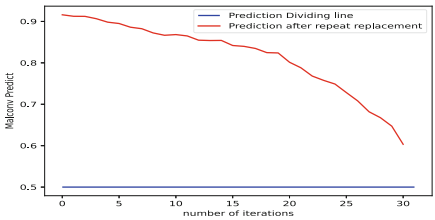



**Fig. 3.** The effect of replacement instructions by repeat replacement.

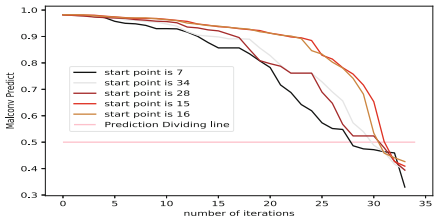



**Fig. 4.** The effect of replacement instructions by continuous replacement.

***Continuous Replacement***. We use "continuous replacement" strategy that replaces each align instructions block with benign instructions sequences that reduce the maximum prediction value to get the locally optimal solution. Taking the same malware as an example, it costs 10,200 iterations to get the result. We randomly select 5 align areas as starting points to verify different starting points has the same effect and then in address order replace align instructions with benign instructions sequences that reduce the maximum prediction value. In Fig. 4, we find "continuous replacement" can decrease the prediction value to 0.5 below every time, but the entire replacement takes 30 min to complete.

### 3.2 Gradient Attack

Goodfellow et al. [6] explain that CNNs are vulnerable for being too "linear" in his article, and introduce the Fast Gradient Sign Method (FGSM) to produce adversarial image samples. The Malconv is a CNN-based detection model, thus we use the FGSM to replace align instructions. The FGSM will take less time than the previous strategies (repeat replacement and continuous replacement) because it just requires one iteration. Different from adversarial image examples, instructions sequences should have concrete program semantics and replacement should follow Portable Executable format. We enhance the FGSM to generate adversarial malware examples. The core of the FGSM is to obtain the gradient loss value of the aim model and amplify the loss to perturb the aim model, making it misclassify. The specific formula is as follows.

$$A' = A + \varepsilon * sign(\nabla_A J(A, B)) \tag{1}$$

where (1) $A$ is the embedding matrix transformed by malware raw byte, $B$ is the null matrix that only contains zero, $\varepsilon$ is perturbation coefficient, *sign()* is to take out the sign of the loss function *J(A,B)*, $A'$ is the perturbation matrix.

The perturbation matrix obtained by the FGSM is not suitable to generate malware examples, since replacement with the perturbation matrix does not follow the Portable Executable format and the data in the perturbation matrix has no concrete program semantics. To solve the problem, we reduce its dimension and use a mapping method that converts the data to x86 instructions. In Algorithm 1, we introduce how to enhance the FGSM to get a noise vector that contains x86 instructions and how to index the align instructions.

In the function EFGSM, we input the original malware file $X_0$ and output the replacement instructions vector $X'$. We first get the null matrix which presents the benign label and the embedding matrix from the Malconv. We input them to FGSM and get the perturbation matrix. To keep the same dimension as the input file, we reduce the perturbation matrix dimension and get a noise vector. We set the x86 instruction *InstructionList* as a mapping table, round and map the data in noise vector to the $X'$. Since the rounding of some data is not in the *InstructionList*, we use the function Getnear to traverse to get the closest instruction. All the mapped instructions are stored in $X'$. By mapping, we can replace align instructions with other instructions, breaking reverse analysis.

In the function AIVG, we first input the PE file $X_0$, then get the address of the .text section, divide the functions and record the address of each function. Each function block is queried to record the offset addresses of all align instructions, and save them in the vector $Y'$. All align instructions in the malware are replaced with instructions that are in $X'$, according to the $Y'$ which contains the offset addresses of all align instructions. The flow chart of the gradient attack strategy is shown in Fig. 5.

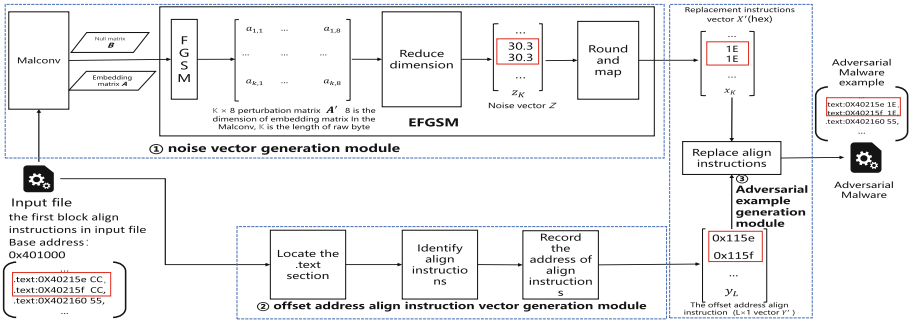



**Fig. 5.** Gradient attack strategy flow chart

**Algorithm 1.** Replace Align instructions with the EFGSM

**Input:** $X_0$: original malware file ;

```
function EFGSM(X_0)
    Nullmatrix, Embeddingmatrix ← Malconv(X_0);
    Perturbationmatrix ← FGSM(Nullmatrix, Embeddingmatrix);
    Noisevector ← Reduce_Dimension(Perturbationmatrix);
    for x in Noisevector do
        if ⌊x⌋ in InstructionList then
            X' ←⌊x⌋;
        else if ⌈x⌉ in InstructionList then
            X' ←⌈x⌉;
        else
            X' ← Getnear(x);
function AIVG(X_0)
    O ← GetTextSection(X_0);
    base ← GetBaseAddress(X_0);
    Y' ← Array(0);
    Func_block ← GetFuncBlock(O);
    for addr_start,addr_end in Func_block do
        i ← addr_start;
        while i < addr_end do
            if X_0[i] is align instruction then
                Y' ← (i − base);
            i + +;
    return Y';
```

**Output:** The adversarial malware;

## 4 Evaluation

We evaluate the superiority of our strategy through comparative experiments and use antiviruses to test the evasion effect of the generated samples.

### 4.1 Dataset and Malconv Setup

Our malware come from virusshare and vx-underground, which provide 100+ new malware families. We collect 200 malware which are come from 11 malware families and 100 benign PE files. All experiments run over Ubuntu16.04 and a system of i7-9700 CPU operating, with 16 GB DDR4 RAM.

### 4.2 Experiment Results

We firstly train a Malconv model using the dataset ember 2018 [9]. Our dataset contains 196 malware and 86 benign files that are correctly predicted. The number of each malware family in our dataset shows in Fig. 6.

In order to verify the accuracy of the EFGSM and prove that it is superior to the previous strategies, we compare the PE head strategy, the PE tail strategy,

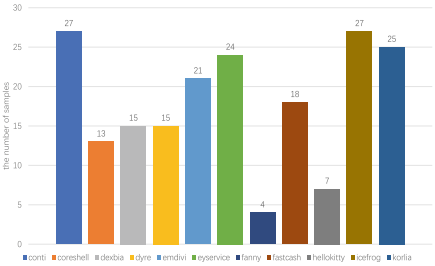



**Fig. 6.** The number of each malware family in our dataset

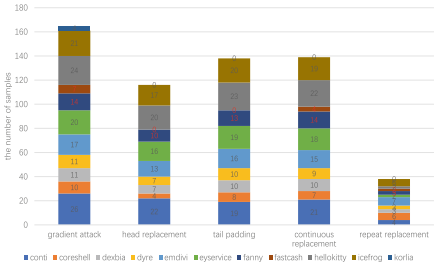



**Fig. 7.** The number of malware that evade the Malconv detection

"repeat replacement" strategy and "continuous replacement" strategy with our gradient attack strategy in the same dataset. 163 adversarial malware samples and 48 adversarial benign samples that the EFGSM generated can make the Malconv misclassification. We analyze samples that don't make the Malconv misclassification. We find that such samples have few align instructions so samples do not have enough instructions to reduce the prediction value below 0.5. We reproduce the PE head strategy that perturbs the DOS headers in 150 iterations to generate adversarial malware samples. Only 116 samples can evade the Malconv. The reason the rate of evasion is low that is our benign byte sequences maybe not be enough. We also reproduce the PE tail strategy that pads benign byte sequences to the PE tail to evade the Malconv. 136 samples can evade the Malconv. We analyze the samples that can not evade the Malconv generated by the PE tail strategy. Their size is more than 2 MB, so the padding strategy may fail. The "repeat replacement" only generate 40 malware that can evade the Malconv and the "continuous replacement" generate 139 malware. The efficiency comparison results are shown in Fig. 7 and the total time to generate 196 examples with every strategy shows in Fig. 8.

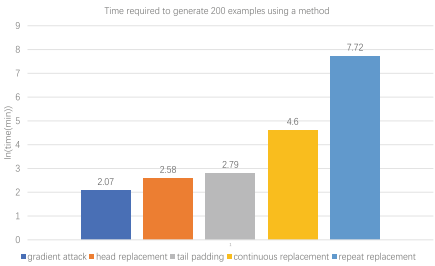



**Fig. 8.** Total time to generate 196 examples

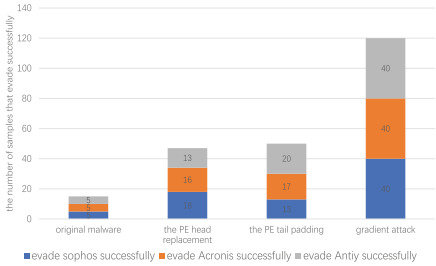



**Fig. 9.** The number of malware that evades antivirus

Modifying PE Header and padding the end of malware for the total 196 samples both take more than 13 min, but our EFGSM only takes 8 min to generate adversarial examples. "Repeat replacement" takes 100 min and "continuous

replacement" takes 2263 min, they come at a considerable penalty in terms of time. By comparing the generation time and the detection rate, our strategy takes less time and has a better effect.

Our goal is not only to evade the Malconv detection but also to further evaluate the effect in the real world. We install locally three anti-virus software (the sophos and the Acronis are based on deep learning detection and the Antiy is a famous static detection), and generated examples with our strategy and previous strategies are tested on them. Figure 9 depicts the total example's evasion effect on antivirus. The number of malware generated by our strategy that can evade one antivirus is 40, but samples generated by other strategies are no more than 20. We speculate modifying the align instructions perturb static detection, so our samples may evade antivirus.

## 5 Related Work

Especially for the Malconv architecture, many attack schemes have been proposed. Kolosnjaji et al. [1] appends benign bytes to the end of malware to evade the Malconv detection. It uses gradient descent to select each byte that is appended to the file. They achieve a 60% evasion rate against the Malconv with 200 samples, a byte appending limit of 10,000, and a max iteration count of 20. Yuan et al. [10] implement a Generative Adversarial Network to generate payloads that are appended to the end of the malware to generate an adversarial malware. Luca et al. [2] insert benign bytes into locations relevant to the header. It was found that from their 60 sample dataset, 52 of these samples can evade the Malconv. Suciu et al. [3] extend the gradient-based approach and systematically evaluated strategies for slack spaces injection and end injection. Their research shows that the strategy of injecting into the slack space that is between PE sections is better than the strategy of injecting into the end of malware. Because of the Malconv's file size limit, the strategy of injecting into the end may fail. Park et al. [11] propose AMAO (Adversarial Malware Alignment Obfuscation), which inserts NOPs into locations of malware. But they get an evasion rate of 100% with an undisclosed amount of samples.

## 6 Conclusion

In the paper, we mainly do research on the align instructions replacement strategy. We find that a tiny replacement can cause the Malconv's prediction value to float. Based on such findings, we enhance FGSM which is not suitable to generate malware samples by mapping data in perturbation matrix to x86 instructions. Conducting experiments and comparisons on the collected 200 malware and 100 benign files, we get a successful evasion rate of 81.5%. Some examples we generate can also evade the detection of anti-virus software.

The Malconv is vulnerable as a CNN model because it is too "linear". Not all information is useful for malware detection, and modifying meaningless information can affect malware detection. Therefore, we propose Malconv add a filtering

layer to retain only valid information, such as file headers, valid instructions in ".text section", ".data section", etc. Doing so can also increase the cost of modifying malware to evade detection. It is also difficult for attackers to add the perturbation matrix into the malware to affect the final detection result. In the follow-up work, we will also do research on other adversarial malware example generation strategies, such as Linux malware and IoT malware.

## References

1. Kolosnjaji, B.: Adversarial malware binaries: evading deep learning for malware detection in executables. In: 2018 26th European Signal Processing Conference (EUSIPCO), pp. 533–537. IEEE (2018)
2. Luca, D., Biggio, B., Giovanni, L., Roli, F., Alessandro, A.: Explaining vulnerabilities of deep learning to adversarial malware binaries. In: 3rd Italian Conference on Cyber Security, ITASEC 2019, vol. 2315 (2019)
3. Suciu, O., Coull, S.E., Johns, J.: Exploring adversarial examples in malware detection. In: 2019 IEEE Security and Privacy Workshops (SPW), pp. 8–14. IEEE (2019)
4. Demetrio, L., Biggio, B., Lagorio, G., Roli, F., Armando, A.: Functionality-preserving black-box optimization of adversarial windows malware. IEEE Trans. Inf. Forensics Secur. **16**, 3469–3478 (2021)
5. Demetrio, L., Coull, S.E., Biggio, B., Lagorio, G., Armando, A., Roli, F.: Adversarial exemples: a survey and experimental evaluation of practical attacks on machine learning for windows malware detection. ACM Trans. Priv. Secur. (TOPS) **24**(4), 1–31 (2021)
6. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
7. Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., Nicholas, C.K.: Malware detection by eating a whole exe. In: Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence (2018)
8. Szegedy, C.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
9. Anderson, H.S., Roth, P.: EMBER: an open dataset for training static PE malware machine learning models. arXiv preprint arXiv:1804.04637 (2018)
10. Yuan, J., Zhou, S., Lin, L., Wang, F., Cui, J.: Black-box adversarial attacks against deep learning based malware binaries detection with GAN. In: ECAI 2020, pp. 2536–2542. IOS Press (2020)
11. Park, D., Khan, H., Yener, B.: Generation & evaluation of adversarial examples for malware obfuscation. In: 2019 18th IEEE International Conference on Machine Learning And Applications (ICMLA), pp. 1283–1290. IEEE (2019)

# Topology Control and Coverage

# Opportunistic Network Routing Algorithm Based on Ferry Node Cluster Active Motion and Collaborative Computing

Gang Xu[1,2(✉)], Qi Tang[1,2], Zhifei Wang[1,2], and Baoqi Huang[1,2]

[1] College of Computer Science, Inner Mongolia University, Hohhot, China
csxugang@imu.edu.cn
[2] Inner Mongolia A.R. Key Laboratory of Wireless Networking and Mobile Computing, Hohhot, China

**Abstract.** In opportunistic network with node clusters, it is usually necessary to set up ferry nodes to connect each cluster to achieve the overall connectivity of the network, and the movement pattern of ferry nodes has an important impact on the overall performance of the network. The existing opportunistic network routing algorithms based on ferry nodes suffer from insufficient resource utilization and inefficient collaboration among multiple nodes in resource optimization and collaborative work, which often leads to low overall network delivery probability and high network load. Therefore, this paper proposes a cooperative routing algorithm for multiple ferry nodes based on active motion mode (ORABAC), in which ferry nodes actively realize the planning of motion paths according to their states and network message forwarding requirements, while multiple ferry nodes in the network realize cooperative work. Simulation results show that the proposed routing algorithm achieves higher delivery probability and less delivery latency while reducing the energy consumption of ferry nodes and restraining network overhead.

**Keywords:** Ferry node · Active motion · Path planning · Collaborative computing · Resource optimization

## 1 Introduction

Opportunistic Network [1] is a kind of wireless mobile ad hoc network that does not have stable links between nodes and relies on node encounters to transmit messages, especially for communication and data collection in remote, disaster or field environments. In an opportunistic network with sparse nodes, the network form disjoint "clusters". It is usually necessary to set up ferry nodes between the fragmented clusters to improve inter-cluster connectivity and help different clusters to complete message transmission. These ferry nodes have larger caches and more energy than other nodes in the network. Since the ferry nodes moving


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 615–626, 2022.
https://doi.org/10.1007/978-3-031-19208-1_51

along a fixed route will frequently move and wait for message transmission in the region without message, it causes a waste of resources of ferry nodes and increases time delay. Therefore, it is important to design ferry node movement rules based on the optimal utilization of resources to improve the delivery ratio and reduce the network overhead.

The existing ferry node-based opportunistic network routing algorithms are mainly: the fixed path-based routing algorithm, the multi-level ferry node collaborative routing algorithms, and the ferry node motion path active planning routing algorithm. The routing algorithm based on a fixed path makes ferry nodes move along the fixed path to complete the communication between nodes. However, in opportunistic network, message generation has characteristics such as uncertainty and unevenness, and accessing node clusters along fixed paths has low utilization of ferry nodes; at the same time, in opportunistic network with uneven message distribution, ferry nodes have high resource consumption and large network delay, which may lead to the existence of urgent messages that cannot be handled effectively and low message delivery probability. The routing algorithm of multi-level ferry nodes usually sets up multiple levels of ferry nodes in the network. All messages in the node cluster are collected by the lower-level ferry nodes, and then the highest-level ferry nodes visit these lower-level ferry nodes along a fixed path, to achieve the delivery of messages. This method reduces the time spent by ferry nodes in clusters to collect messages and can reduce the network latency to a certain extent, but the top-level ferry nodes still move along the fixed path, which does not solve the problem of poor utilization of ferry nodes in the opportunistic network with uneven message distribution. In the routing algorithm based on the active planning of ferry nodes' motion path, ferry nodes dynamically adjust their motion trajectory according to their state and the message forwarding demand in the network, which effectively solves the problems of low utilization of ferry nodes' resources and timely message delivery caused by the motion along the fixed path. But the current research in this field mainly focuses on the case where only one ferry node and little consideration is given to the cooperative work under multiple ferry nodes.

The innovation of this paper is as follows. This paper combines the state of ferry nodes and context information to determine the mobile path, and supports multiple ferry nodes to work at the same time, which overcomes the problem of low utilization of ferry nodes in fixed path routing algorithm and improves the efficiency of cooperative work of multiple ferry nodes.

## 2 Related Work

The routing algorithms for ferry nodes are mainly divided into three types of routing algorithms based on fixed paths [2–5], on hierarchical ferry nodes [6–10], and on active planning of ferry node motion paths [11–16]. However, the existing studies do not apply to clustering routing algorithms with ferry nodes.

In fixed-path-based routing algorithms, ferry nodes realize the communication between clusters in the network by moving along a fixed path. Therefore,

limited to fixed paths, how the ferry node efficiently forward messages becomes particularly important. Zhao, G et al. proposed FTFPF algorithm [2], which gathered messages to the nodes with the highest forwarding tendency in the cluster, got them by ferry nodes, and forwards them between clusters. This algorithm maximizes the single visit revenue of node clusters, but low utilization ratio. Liu, C et al. proposed in CBSW routing algorithm [3] let some ordinary nodes through the cluster center, improved the chance of meeting between ordinary nodes and ferry nodes, and reduced the time of waiting for messages on some ordinary nodes, but the message coverage area of ferry node is small. So OFR routing algorithm [4] was proposed, which allowed ferry nodes to adjust the communication range adaptively. Xue, L. et al. proposed in ADMF [5] to elect temporary ferry nodes from ordinary nodes to participate in network communication according to network load. The algorithm solves the performance limitation of using a single ferry node, but the limitation caused by the fixed path movement still exists and does not consider the problem of repeated access to clusters by multiple ferry nodes.

A hierarchical ferry node strategy can solve the problem of insufficient message coverage caused by a single access cluster of fixed path routing algorithms, but the excessive dependence of this type of strategy on a global ferry node is easy to cause network congestion and even paralysis [6,7]. The strategy mainly uses message aggregation, which is divided into local ferry nodes and global ferry nodes. Find a node in each cluster as a local ferry node to collect messages in the current region. For example, Niitsu, Y proposed to use a regional central node as a local ferry node [8], and then forward messages from the regional ferry node to the global ferry node [9,10]. Ferry nodes at different levels cooperate to complete the global message forwarding of the network, which limits the energy waste caused by the invalid message forwarding between ordinary nodes. However, setting only one global Ferry node under high network load cannot meet the network requirements.

To make full use of the resources of ferry nodes, some scholars proposed to actively plan the ferry node path according to the context information [11,12]. Context information includes access distance, message number, TTL, etc. However, there is a waste of resources when multiple ferry nodes visit the same node region according to the same route. Due to the sociality of nodes, Chen, W et al. [13] defined the relationship strength to plan the movement route of ferry nodes. However, measuring the relationship strength based on historical encounter information is only applicable to the network with strong social relations, and is not suitable for the network with completely random movement of nodes. To maximize the energy utilization of ferry nodes and prolong the network service time, the MPBF routing algorithm [14] was proposed, which considered the energy consumption of ferry nodes. In [15], the selection of cluster heads was based on their residual energies and their distances to the ferry path. This algorithm can effectively reduce the message delivery latency, but in the case of serious network isolation, it will cause the growth of the motion path of the ferry node. To achieve higher data arrival number and hop count reduction, Kazuma

Ikenoue et al. [16] proposed a new message ferry method that enables each node to establish a route to increase the number of data arrivals by shortening paths, but this algorithm lacks consideration of ferry's state.

In summary, the existing routing algorithm based on the active planning of the ferry node motion path is more suitable for the opportunistic network of network node partition, but the delivery ratio is still low when the nodes in the cluster move randomly and the network isolation is serious, and multiple ferry nodes cannot work together efficiently. To solve the above problems, this paper proposes a routing algorithm using the ferry nodes' state and message passing requirements as evaluation indexes to evaluate the utility value of ferry nodes for each node cluster, and dynamically plan the motion path of ferry nodes according to the utility value.

## 3 Routing Arithmetic Design

To realize the purpose of ferry node planning the motion path actively according to its state and the communication requirements in the network, ORABAC is proposed in this paper, which combines the energy, historical encounter, distance, message number, TTL and other dimensions to calculate the utility value of ferry nodes. This utility value can evaluate the rationality of the planned route, so that the planned route can avoid the repeated and inefficient use of the ferry node communication resources, to optimize the network performance. For the whole network, when multiple ferry nodes work together, the node with more energy left is preferred for message forwarding, which can balance the energy consumption of ferry nodes, and prevent these nodes from dropping out of the network service due to excessive energy consumption, resulting in network performance degradation. For a single ferry node, choosing a node cluster with a strong relationship or with more messages as the destination of the next hop can maximize the benefits of single delivery; the nodes with closer access distance can enhance the coverage of ferry nodes on the network without consuming too much energy; At the same time, selecting a node cluster that travels to a smaller message TTL can avoid messages being discarded due to timeout, thereby increasing the network delivery probability.

### 3.1 Relevant Definitions

These related indicators used to calculate the utility value of ferry nodes are defined as follows:

**Definition 1.** *Node Residual Energy Ratio*

*In the process of message delivery, whether the residual energy of the ferry node is sufficient is the key to whether the message can be delivered. Therefore, when selecting the ferry node, the energy should be considered. In this paper, the node residual energy ratio is used as the judgment basis of the node energy state.* $E_n(t)$ *represents the node residual energy ratio of node N at t time, and its definition is shown as Eq. (1).*

$$E_n(t) = \frac{E_{cut}(t)}{E_{max}} \tag{1}$$

We use $E_{max}$ to represent the maximum energy stored by node N, and use $E_{cut}(t)$ to represent the residual energy of node N at time t.

**Definition 2.** *Node Relationship Strength*

*The encounter frequency between nodes can be used to indicate the degree of closeness between them. The more frequent encounters, the more frequent information communication between nodes, that is, closer relationships. Ferry nodes also have frequently visited regions and not frequently visited regions. The frequency of encounters between ferry nodes and nodes in frequently visited regions is often higher, and the messages carried by ferry nodes to this region will also be more. Handing these messages to ferry nodes with closer relations can complete the delivery, and more messages can be delivered when the node moves to this region, to maximize the use of the energy of node movement.*

*This paper reflects the social relationship between nodes according to the number of historical encounters between nodes. Within the time length t, the relationship strength between node i and node j is expressed as $F_{(i,j)}(t)$, and the calculation method is shown in Eq. (2):*

$$F_{(i,j)}(t) = \frac{E_{(i,j)}(t)}{N_i(t)} \tag{2}$$

where $E_{(i,j)}(t)$ represents the number of encounters between node i and node j within time length t, and $N_i(t)$ represents the total number of encounters between node i and all its neighbors within time length t.

**Definition 3.** *Distance Between Node and Message Source Node*

*When selecting the ferry node to actively move to the vicinity of the message source node for message reception, the closer the distance $D_i$ between the ferry node and the message source node is, the faster the ferry node receives the message and the shorter the time it takes to complete the message delivery. In this paper, the normalized distance between the message source node i and the ferry node is used as the distance metric between nodes, and the calculation method as Eq. (3) shows:*

$$D_i = \frac{d_i}{\sum_{i=1}^{n} d_i} \tag{3}$$

$$d_i = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2} \tag{4}$$

In Eq. (3), n is the number of nodes in the network, and $d_i$ is the Euclidean distance between the ferry node and the message source node i. The calculation method is shown in Eq. (4), where $x_f$ and $y_f$ are the positions of the ferry node, and $x_i$ and $y_i$ are the positions of the message source node i.

**Definition 4.** *Normalized Message Number*

*The more messages ferry nodes carry to a certain node cluster, the longer the total waiting time of messages, and the greater the probability of the cluster being selected as the next hop. Designing a ferry node to move to the cluster with more messages can increase the number of successful messages delivered and reduce the waiting time of messages, thereby increasing the delivery probability of the network and reducing the average delay of the network. Defines $M_i$ as the normalized number of messages of all destination addresses carried by the ferry node as messages of nodes in node region i, as shown in Eq. (5):*

$$M_i = \frac{m_i}{\sum_{i=1}^{n} m_i} \tag{5}$$

In the expression, $m_i$ is the total number of messages carried by the current ferry node to the node in cluster i, and n is the number of clusters.

**Definition 5.** *Normalized Message Waiting Time Simply considering the number of nodes may lead to the abandonment of messages with smaller TTL. Therefore, the method in this paper gives priority to sending TTL smaller messages when planning the motion path. When the number of messages carried by ferry nodes to each cluster is more than one, we judge the cluster with a smaller total message waiting time by Definition 5. $t_i$ is defined as the sum of messages TTL of all destination nodes carried by nodes in cluster i, and the calculation method is as follows:*

$$t_i = \sum_{m=1}^{n} TTL_m \tag{6}$$

where n is the number of messages in the destination node cluster i and $TTL_m$ is the TTL value of message m.

$T_i$ is defined as the normalized message waiting time of cluster i, which is calculated as follows:

$$T_i = \frac{t_i}{\sum_{j=1}^{n} t_j} \tag{7}$$

where n is the total number of clusters in the network.

### 3.2 An Active Path Planning Routing Algorithm for Multi-ferry Nodes

To select the ferry node with the best overall network performance from multiple ferry nodes for forwarding, the ORABAC algorithm proposes the utility value to measure the forwarding message benefit of the ferry node based on the contextual information such as remaining energy, relationship strength, node distance, message number and message normalization waiting time. The ferry node to cluster i is evaluated as shown in Eq. (8)

$$V_i = -lb\frac{E_n(t) \times F_{(i,j)}(t) \times M_i \times T_i}{D_i} \tag{8}$$

When a node has a message to send, it sends MR information to the surrounding by broadcasting while suspending its motion. MR information is used to request ferry node access messages, which include the location of the current node, the destination node of the message, and the message to be transmitted. The ferry node within the scope receives the broadcast MR information first to determine whether it has carried the message, and if not, calculates its utility value $V_i$ for the message through Eq. (8) based on the data in the MR information, and also transmits the message back to the source node through broadcast.

The source node receives the utility value of the Ferry node and selects the Ferry node with the largest value as the message delivery service. The selected ferry node actively moves to the location of the message source node and obtains the message that needs to be forwarded and recalculates the node cluster to be accessed in the next step after obtaining the message, or continues to move according to the original plan. After the ferry node reaches the cluster where the destination node is located, all messages carried by the destination node for the current node in the cluster are transmitted to the encountered node, and then these messages are continuously forwarded in the cluster through flooding until delivery is completed.

## 4 Experimental Simulation and Result Analysis

### 4.1 Experimental Parameters and Scene Settings

This paper uses The ONE 1.4.1 to verify the proposed algorithm. In this paper, simulation experiments are designed for different scenes such as sparse distribution of nodes and dense distribution of nodes. The performance of the ORABAC algorithm proposed in this paper is analyzed and the effects of the ORABAC algorithm, ERMF [10] algorithm, and FTFPF [2] algorithm are compared. The parameter settings in the ONE are shown in Table 1.

**Table 1.** Simulation experiment parameter settings

| Parameter | Value |
|---|---|
| Experimental area size | 6450 × 5340 |
| Number of node clusters | 4–6 |
| Node mobility model in cluster | RandomWayPoint |
| Number of nodes in cluster | 5–50 |
| Moving speed of cluster nodes | 3.5–5.5 m/s |
| Number of nodes between clusters | 1–7 |
| Mobile model of interregional nodes | MapRouteMovement |
| Speed of interregional nodes | 18.5–20.5 m/s |
| Communication radius of high-speed interface | 50 m |
| Communication radius of low speed interface | 2000 m |
| Experimental simulation time | 24 h |

The simulation scenes are shown in Fig. 1, where the number of node clusters and the closeness of their distribution vary from scene to scene. The nodes in the cluster move randomly, and single or multiple ferry nodes are set between the clusters. The specific scenes are as follows: 1) scene 1: The operation environment with sparse node distribution in practical application is simulated, such as the application of opportunistic network in grassland, village, remote agricultural, and pastoral areas. In this scene, the number of node clusters is small, and the distribution of node clusters in geography is also sparse. There are four clusters, and 1–5 ferry nodes are set between node clusters. 2) scene 2: Simulation of the operating environment in which nodes are densely distributed in practical applications, such as campus and urban opportunistic network node deployment. Compared with scene 1, the scene has more node clusters and the distribution of node clusters is more intensive, which contains 6 clusters, and there are 1, 3, 5, 7 ferry nodes among the node clusters.

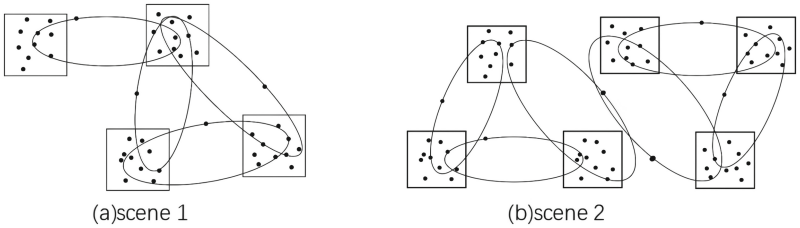



**Fig. 1.** Simulation environment.

## 4.2 Experimental Results and Analysis

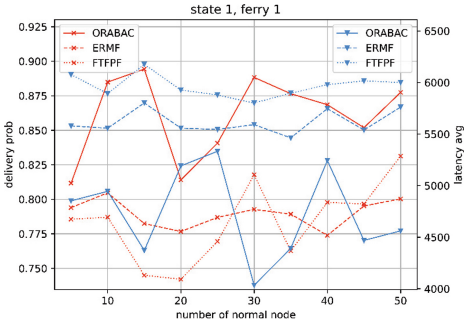



**Fig. 2.** Performance comparison of single ferry node.

Figure 2 and Fig. 3 are the experimental results of the delivery probability and latency of different ferry nodes in the scene of sparse node cluster distribution.

The left ordinate axis represents the message delivery probability, and the right ordinate axis is the average delay. Figure 2 is the delivery probability of the message and the average latency of the three routing algorithms when only one ferry node is set in the network. Figure 3 is the comparison of network performance indicators when 2–5 ferry nodes are set in the network. By comparing Fig. 2 and Fig. 3, it can be seen that the delivery success ratio of the ORABAC routing algorithm proposed in this paper is significantly higher than that of ERMF and FTFPF algorithms, and the latency average is significantly smaller, no matter whether a single ferry node or multiple ferry nodes are set in the network. Therefore, the ORABAC routing algorithm has good applicability in the case of sparse distribution of node clusters in the network. It can also be found from the figures that in most cases, the delivery probability of ERMF is similar to that of FTFPF. This is because the routing algorithm based on the hierarchical ferry node is still essentially moving along a fixed path. The problem of resource waste in FTFPF, a routing algorithm based on a fixed path, has not been well solved, so the delivery probability is not greatly improved. The proposed active path planning method can effectively solve the problem of resource waste, so the delivery probability is greatly improved compared with the other two algorithms.

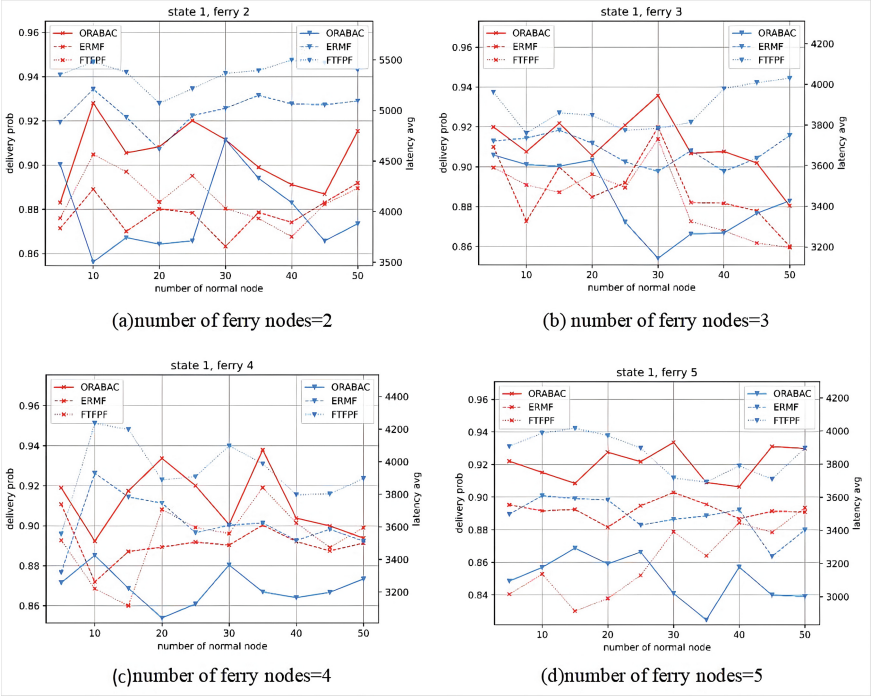



**Fig. 3.** Performance comparison of multiple ferry nodes

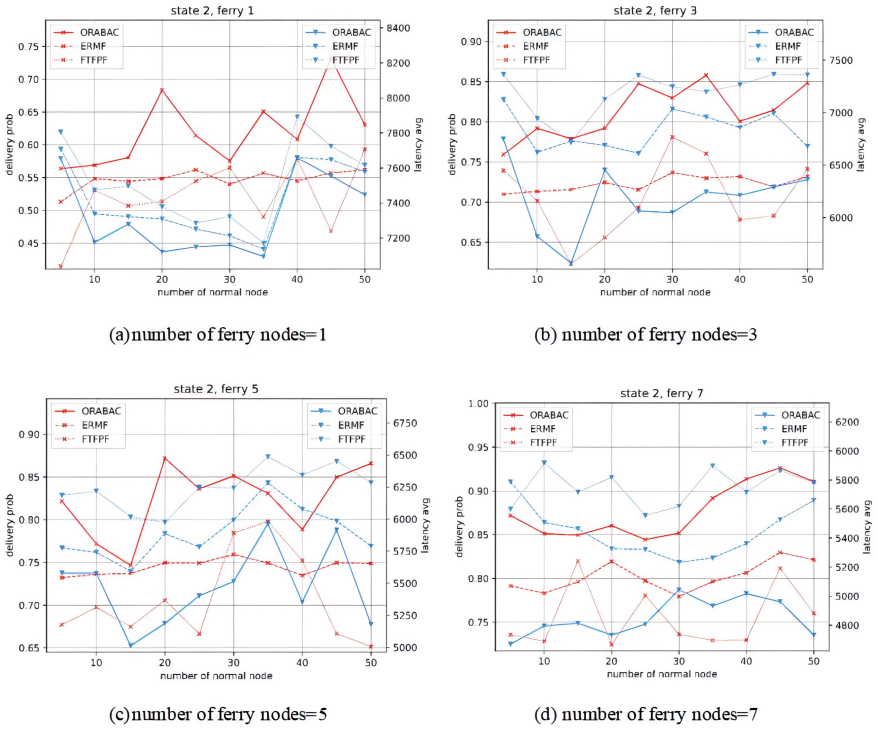



**Fig. 4.** Performance comparison of multiple ferry nodes

Figure 4 is the experimental results of different routing algorithms in scene 2 with different numbers of ferry nodes. Figure 4(a)is the algorithm comparison graph of setting a single ferry node in the network. Figure 4(b), 4(c) and 4(d) are the network performance comparison graphs when 3, 5, 7 ferry nodes are set in the network. It can be seen from the figure that in scene 2, the delivery probability of the ERMF algorithm and FTFPF algorithm is lower than that of the ORABAC algorithm, while their latency is much higher than that of the ORABAC algorithm, indicating that the algorithm proposed in this paper can effectively improve the delivery probability and reduce the network overhead. By analyzing and comparing the latency of three algorithms, it can be found that the ORABAC algorithm is far less than the ERMF algorithm based on node stratification. This is because the ORABAC algorithm analyzes the carrying time of node messages in path planning, and preferentially transmits messages with a long waiting time.

To sum up, the routing algorithm proposed in this paper has good performance in both scenes, which can improve the delivery probability while significantly reducing latency, effectively improving network performance, and also has good support for multiple ferry nodes to work together.

## 5 Conclusion

This paper proposes an opportunistic network routing algorithm based on cooperative computing active mobility of ferry nodes, which solves the problem that ferry nodes exit the network due to insufficient network resources and the low efficiency of inter-cluster cooperation. This algorithm combines the residual energy ratio of the ferry node, the strength of the relationship between the encounter state of the nodes, the Euclidean distance between the ferry node and the source node, the number of messages carried on the ferry node, and the normalized waiting time of the messages carried to propose the utility value of the ferry node, which is used as the judgment basis to select the appropriate ferry node as the carrier for message transmission. It solves the problems of uneven energy consumption of nodes in the two routing algorithms based on the fixed path motion algorithm and the ferry node hierarchical algorithm and frequent message congestion when there are too many messages. It effectively improves delivery probability, reduces the network overhead, balances the energy consumption speed of each ferry node, and greatly prolongs the network lifetime.

**Acknowledgment.** This work was partially supported by the National Natural Science Foundation of China under Grant 62061036,61841109 and 62077032, Natural Science Foundation of Inner Mongolia under Grand 2019MS06031, Inner Mongolia Autonomous Region Graduate Research Innovation Project S20210127Z.

## References

1. Soelistijanto, B., Howarth, M.: Transfer reliability and congestion control strategies in opportunistic networks. IEEE Commun. Surv. Tutorials **16**(1), 538–555 (2014)
2. Zhao, G.-S., Chen, M.: Forward tendency based fixed path ferry routing algorithm. J. Beijing Univ. Posts Telecommun. **35**(2), 41–45 (2012)
3. Liu, C.-R., Zhang, S.-K., Jia, J.-C., Lin, C.-K.: Routing mechanism based on the cooperation of the ferry nodes and cluster nodes in opportunistic networks. Acta Electron. Sin. **44**(11), 2607–2617 (2016)
4. Peng, C., Li, W.H., Wang, Y.Z.: All coverage and low-delay routing algorithm based on message ferry in opportunistic networks. Appl. Res. Comput. **34**(03), 819–823 (2017)
5. Xue, L., Liu, J., Peng, J.: An adaptive message ferry routing algorithm for Delay Tolerant Networks. In: 2012 IEEE 14th International Conference on Communication Technology CONFERENCE 2012, pp. 699–703. Institute of Electrical and Electronics Engineers Inc., Chengdu (2012)
6. Tang, L.J., Chai, Y., Li, Y.: Route design for multiple message ferries in partitioned opportunistic networks. Appl. Res. Comput. **30**(06), 1775–1778 (2013)
7. Li, Y., Weng, B.B., Liu, Q.L.: Multiple ferry route design based on city-village model in opportunistic networks. Appl. Res. Comput. **029**(1), 263–265 (2012)
8. Xiong, X.R., Zhang, N., Ji, R.J.: Routing strategy of regional center node in post-disaster delay tolerant network. Comput. Eng. Des. **40**(06), 1529–1534 (2019)
9. Niitsu, Y., Sakuma, T., Date, H.: Power utilization efficiency improvement method for DTN using a message ferry. In: 8th International Conference on Ubiquitous and Future Networks Conference 2016, pp. 954–956. IEEE Computer Society, Vienna (2016)

10. Li, J.B., Deng, K., Ren, Z.: An efficient and low-delay routing algorithm for multiple ferries in opportunistic networks. J. Xi'an Jiaotong Univ. **49**(04), 91–97 (2015)
11. Roy, S., Bhusal, S., Tomasi, D., et al.: Optimizing message ferry scheduling in a DTN. In: 16th ACM International Symposium on Mobility Management and Wireless Access, pp. 113–117. Association for Computing Machinery Inc., Montreal (2018)
12. Alaoui, E.A.A., Amine, K., Moudden, M.E., Agoujil, S.: Towards an efficient circulation of message ferry in the DRHT. In: Proceedings of the 3rd International Conference on Smart City Applications Article, vol. 30. Association for Computing Machinery, Tetouan (2018)
13. Chen, W., Chen, Z., Li, W., Zeng, F.: An enhanced community-based routing with ferry in opportunistic networks. In: 2016 International Conference on Identification. Information and Knowledge in the Internet of Things, January 2018, pp. 340–344. Institute of Electrical and Electronics Engineers Inc., Beijing (2016)
14. Vallikannu, R., George, A., Srivatsa, S.K.: Routing and charging scheme with ferry nodes in Mobile Adhoc networks. In: 2017 International Conference on Intelligent Computing and Control, I2C2 2017, 23 June 2017–24 June 2017, 1–4 January 2018. Institute of Electrical and Electronics Engineers Inc., Coimbatore (2017)
15. Alnuaimi, M., Shuaib, K., Alnuaimi, K., Abdel-Hafez, M.: Ferry-based data gathering in wireless sensor networks with path selection. Procedia Comput. Sci. **52**(1) (2015)
16. Ikenoue, K., Ueda, K.: Routing method based on data transfer path in DTN environments. In: Barolli, L., Hellinckx, P., Enokido, T. (eds.) BWCCA 2019. LNNS, vol. 97, pp. 544–552. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-33506-9_49

# Optimal Deployment and Scheduling of a Mobile Charging Station in the Internet of Electric Vehicles

Zhenxian Ma[1,2(✉)], Ran Wang[1,2], Changyan Yi[1,2], and Kun Zhu[1,2]

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
{mzx,wangran,changyan.yi,zhukun}@nuaa.edu.cn
[2] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China

**Abstract.** As an alternative to traditional vehicles, electric vehicles (EVs) have significantly increased their market share in recent years. However, the limited battery capacity of EVs may become a bottleneck in their development. Mobile charging vehicles (MCVs), as emerging charging devices, can provide a more portable charging mode. The MCV deployment and charging strategy plays a decisive role in the effective operation of the whole Internet of Electric Vehicles (IoEV). In this paper, we investigate the joint MCV deployment and charging schedule (JMDCS) in one-to-many mode. An integer linear programming problem is formulated to minimize the completion time, consisting of the deployment time of the MCVs and the time of the charging schedule. Since this problem is NP-hard, an approximate algorithm is proposed, where a special rounding technique is employed to assign jobs to the plugs. The simulation results show that when the number of EVs is small, the proposed algorithm is close to the optimal algorithm. When the number of EVs is large, the proposed algorithm outperforms its counterparts in scheduling performance and shows superiority over the optimal algorithm in scheduling efficiency.

**Keywords:** Mobile charging vehicles (MCVs) · Internet of electric vehicles (IoEVs) · Rounding technique

## 1 Introduction

In recent years, one goal has been to use electricity to replace oil, and as a new type of energy, electricity has been applied to automobiles. Electric vehicles (EVs) are more environmentally friendly than traditional petroleum vehicles. However, the limited electric capacity of EVs hampers their development so that they cannot be applied on a large scale. Compared with gasoline vehicles,

This work is supported by the National Natural Science Foundation of China under grant No. 62171218.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 627–639, 2022.
https://doi.org/10.1007/978-3-031-19208-1_52

EVs require a longer charging time and have a shorter cruising range. Therefore, EV charging is a great challenge. For this challenge, a variety of solutions have been proposed, such as battery swapping and fixed charging stations (FCSs).

Swapping batteries is a large economic investment, and it can also lead to much battery waste. In addition, the battery models of different EVs are not necessarily compatible, so it may cause damage to the EVs as well as much waste; FCSs can alleviate the battery anxiety of EV users to a certain extent, but the fixed locations of FCSs will make it impossible to provide charging services for EVs that are randomly distributed, especially in remote areas. During a low charging period, a large number of charging piles are idle, but during a peak charging period, a large number of charging requests cannot be satisfied. Intensive construction of FCSs in a city will inevitably lead to wasted space. For users who need emergency charging, FCSs seem unsatisfactory.

To address the above challenges, in the Internet of Electric Vehicles (IoEV), mobile charging stations (MCSs) provide an efficient charging alternative: by utilizing mobile charging vehicles (MCVs) as an alternative mobile charing method. Compared with battery swapping, MCVs do not cause much waste on the facility utilization, and they do not require much capital investment and only provide charging services. In addition, MCVs are more flexible than FCSs and can provide charging services at any place and any time without requiring much floor space. For users who need emergency rescue, MCVs can be quickly deployed to designated service locations.

The MCV deployment and charging strategy becomes the key to providing charging services for EVs. In this paper, we consider a situation in which emergency rescue is provided for EVs and users in a certain area initiate charging requests. The MCV is equipped with multiple plugs that can provide one-to-many simultaneous charging. When users in an area initiate a request, the MCV selects a deployable location in the area (such as a parking lot) to provide services for the requesting users. When the MCV selects a deployment location, users who need charging can move to a small area in the parking lot that the initial power of each EV is guaranteed to reach. The MCV decides which plug to assign the users to in order to achieve the goal of completing all charging tasks as quickly as possible. The MCV is concerned with two aspects of the scenario: the deployment site and the plug assignment strategy. In general, since a certain amount of energy will be consumed when an EV moves to the designated location, resulting in changes in the charging time, the selection of the deployment location will affect the allocation strategy. In this paper, we formulate this problem of deployment and assignment as an integer linear programming problem. The main contributions of our work are as follows:

- To address the challenges of battery swapping and FCSs, we propose the concept of mobile charging. The MCV is equipped with multiple charging plugs, which can provide simultaneous one-to-many charging.
- We model the deployment problem and the charging schedule problem as an integer linear programming problem. Since this is an NP-hard problem, we propose an approximate algorithm to solve it.

– Simulations are carried out to verify the performance efficiency of our algorithm and other algorithms. The results show that our approximation algorithm outperforms the other algorithms on small-scale problems but does not achieve optimal results. On large-scale problems, we cannot obtain optimal results, but the approximate solution obtained by the proposed algorithm has a large advantage over those of the other algorithms.

The remainder of this paper is structured as follows: The related work is given in Sect. 2. Section 3 describes the system model. In Sect. 4, we introduce the formulation of our problem. Section 5 proposes an approximation algorithm. The numerical results and analysis are presented in Sect. 6. Finally, conclusions are drawn in Sect. 7.

## 2 Related Work

In this section, we discuss works that exploit MCVs for charging EVs. In the literature, much work revolves around how the MCV determines the next charging target. Huang *et al.* in [1] proposed a mobile charging service in an urban environment, and then a queuing-based model was used to determine the scope of the MCV implementation. A nearest-job-next (NJN) service strategy was adopted to determine the next EV to serve. Liu *et al.* in [2] proposed an assignment rescheduling mechanism that replans the service object of the MCV to reduce the charging expenses of EVs. Chen *et al.* in [3] considered the dynamic arrival of EVs and aimed to maximize the long-term average profits of MCVs; a stochastic optimization problem was proposed to determine the scheduling of MCVs. Finally, based on Lyapunov optimization theory, a Lyapunov-based online distributed algorithm was proposed to obtain the optimal solutions. In [4], Zhang *et al.* used neural networks to predict charging needs in cities. Based on this, the charging pressure in a region is relieved by externally dispatching MCVs. Finally, an optimal service location is found through teaching-learning-based optimization. Wang *et al.* in [5] proposed a multi-objective MCV scheduling problem aiming to maximize the average charging benefits of all EVs and minimize the average waiting time of all EVs. Deep reinforcement learning was then used to find the Pareto fronts for multi-objective optimization. All the above studies concern the scheduling problem of mobile charging vehicles finding a charging sequence through different methods. However, they only provide one-to-one services, which is not efficient in the IoEV.

In the field of mobile charging, one-to-many services must be more efficient than one-to-one. When charging wireless sensor networks, one-to-many charging mode is mostly used. Ma *et al.* in [6] proposed a one-to-many charging scheme that allows multiple sensors to be charged simultaneously by a single charger. Through trajectory planning of the MCV, the sensor energy expiration time was minimized. In [7], Xu *et al.* formulated a novel longest charging delay minimization problem to address fair scheduling in one-to-many charging mode. Based on this, we can study the use of one-to-many charging methods when charging EVs.

## 3 System Model

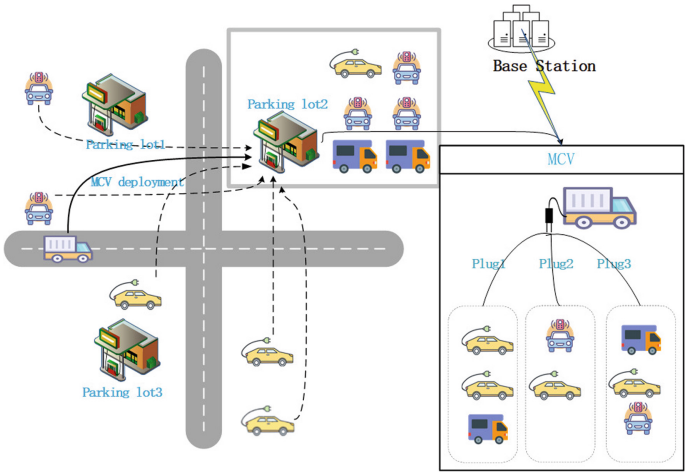



**Fig. 1.** Illustration of the IoEV network model.

In this paper, we consider an IoEV network model consisting of one MCV and $N$ electric vehicles (EVs) denoted by the set $N = \{e_1, e_2, ..., e_n\}$ distributed in a two-dimensional plane, as shown in Fig. 1. Each vehicle is equipped with a sensor to monitor the state of charge (SOC) in the battery and collect data. When the SOCs of the EVs are low, the EV users send requests to the base station, and then an MCV is dispatched to a parking lot. In addition, there are multiple plugs on the MCV. Let $F = \{f_1, f_2, ..., f_m\}$ denote the plugs of the MCV. With multiple plugs in an MCV, the simultaneous charging needs of multiple vehicles can be met. Since it is necessary to complete all charging tasks in the area and eliminate repeated scheduling of the MCV, we consider gathering nearby EVs into one parking lot. The MCV needs to find an existing parking lot (such as parking lot 2 in Fig. 1) and assign EVs to plugs reasonably to minimize the total time consumption.

When the MCV is fully charged, the power consumption of the MCV itself is ignored. In the case of emergency charging, the EVs send a charging request to the base station. The base station calculates the deployment parking lot of the MCV after processing, and the EVs move to the parking lot to wait for charging. In this process, each EV ensures that the remaining power enables it to move to the selected parking lot. We consider Cartesian coordinates and denote the positions of MCV $m$ and EV $i$ as $r_m = (x_m, y_m)$ and $r_i^{ev} = (x_i^{ev}, y_i^{ev}), i = 1, 2, ..., n$, respectively. There are several different parking lots located in this area, denoted by the set $L = \{l_1, l_2, ..., l_q\}$, and their coordinates are $r_q^l = (x_q^l, y_q^l)$. When an EV moves to the selected parking lot, energy loss will occur, which will inevitably

affect the subsequent charging time. Next, we define the energy consumption model of the EVs.

During the movement of an EV to a parking lot, the energy consumption is denoted as:

$$C_i^{ev} = \alpha d_{des}^i, \tag{1}$$

where $\alpha$ is the consumption per kilometer and $d_{des}^i = \|r_i^{ev} - r_{des}\|_2$ is the distance between EV $e_i$ and the selected parking lot. We use $t$ to represent the time consumption of the MCV,

$$t = \frac{\|r_m - r_{des}\|_2}{v}, \tag{2}$$

where $v$ is the speed of the MCV. Since the EVs are close to the parking lot, we assume that the EVs arrive before the MCV. Therefore, the total time consumption of MCV deployment is $t$.

After all EVs arrive at the parking lot, the power of an EV is $E^{init} - C_i^{ev}$, where $E^{init}$ is the initial quantity of electricity of the EV. The MCV is equipped with $m$ plugs, and they work in the same way. We consider our scheduling problem, where there is a fixed charging time $p_{ij}$ associated with each plug $f_j, j = 1, 2, .., m$ and each $e_i, i = 1, 2, ..., n$. The total charging time of each plug is $T_j$. Our goal is to make every plug charging time as close to the average possible and to complete all tasks in the shortest time. During movement, the consumption of each EV is $C_i^{ev}$, the residual energy is $E^{init} - C_i^{ev}$, and the energy required by $e_i$ is $E_i^d = E_{max} - E_i^{init} + C_i^{ev}$. The charging time $p_{ij}$ of each EV is

$$p_{ij} = \frac{E_i^d}{\gamma}, \tag{3}$$

where $\gamma$ is the charging rate.

## 4 Problem Formulation

In this section, we describe the formulation of the deployment problem and charging schedule. Because one of the $Q$ parking lots needs to be selected as the destination, we use $\tau_q \in \{0, 1\}$ to represent whether parking lot $q$ is selected; if $\tau_q = 1$, then parking lot $q$ is selected; otherwise, it is not. Vector $F' = \{\tau_1, \tau_2, ..., \tau_Q\}$ is the deployment vector. To ensure that only one location is selected, we have the constraint that $\sum_{q=0}^{Q} \tau_q = 1$.

Considering the relationships between EVs and the charging plugs, the following $0 - 1$ variables are defined:

$$x_{ij} = \begin{cases} 1, & \text{if EV}_i \text{ assigned on plug } j; \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The total time of plug $j$ is $T_j = \sum_{i=1}^{n} x_{ij} p_{ij}$. The maximum processing time of the plugs is $T_{max} = max(T_1, T_2, ..., T_m)$. Therefore, the total time model is given by

$$\boldsymbol{P_0}: \quad \min_{\tau_q, x_{ij}} \quad t + T_{max} \tag{5}$$

$$\text{s.t.} \quad C_i^{ev} \leq E_i^{init}, i = 1, 2, ..., n; \tag{6}$$

$$\sum_{i=1}^{n} x_{ij} p_{ij} \leq T_{max}; \tag{7}$$

$$\sum_{j=1}^{m} x_{ij} = 1, \ i = 1, 2, ..., n; \tag{8}$$

$$x_{ij} \in \{0, 1\}; \tag{9}$$

$$\sum_{q=0}^{Q} \tau_q = 1; \tag{10}$$

$$\tau_q \in \{0, 1\}. \tag{11}$$

Here, constraint (6) states that the amount of electricity consumed is less than the initial capacity of each EV during EV movement. Constraint (7) ensures that the overall charging time of each plug is less than the maximum time. Constraint (8) indicates that each EV can only be assigned to one plug. Constraint (10) means that only one of the parking lots is selected. Constraints (9) and (11) indicate that these are 0–1 variables. Our objective function is to minimize the time for the MCV to complete all tasks.

## 5 Solutions

In this section, we decompose the problem into an unrelated parallel problem (UPM) [8] and then relax the 0–1 binary variable to a continuous variable to obtain a set of continuous solutions. The solution of $\boldsymbol{P0}$ determines the assignments of all EVs to their corresponding plugs. However, the actions regarding assignment decisions should have integer values. We adopt a rounding technique to achieve an approximate solution.

### 5.1 Problem Decomposition

First, in the overall charging process, a parking lot needs to be selected as the destination of the MCV, and then the charging service can be performed. After the parking lot is selected, the MCV and EVs move to the destination for charging. When the location is selected, the battery consumption of each EV in moving to the destination will change, resulting in a change in the charge level $p_{ij}$. In addition, the time needed for the MCV to move to different parking lots will be different, which will affect $t$ in $\boldsymbol{P_0}$. The change between the two will directly affect our final objective function.

We iterate over a small number of parking lots. Every time a parking lot is calculated, the process can be considered to have a fixed $t$ and $p_{ij}$, and the calculation in the deployment phase in $\boldsymbol{P_0}$ can be eliminated.

Therefore, $\boldsymbol{P_0}$ can be transformed into an unconstrained minimization problem (UMP) as follows:

$$\boldsymbol{P_1}: \min_{x_{ij}} \quad T_{max} \tag{12}$$

$$\text{s.t.} \quad C_i^{ev} \leq E_i^{init}, i = 1, 2, ..., n, \tag{13}$$

$$\sum_{i=1}^{n} x_{ij} p_{ij} \leq T_{max}, \tag{14}$$

$$\sum_{j=1}^{m} x_{ij} = 1, \ i = 1, 2, ..., n, \tag{15}$$

$$x_{ij} \in \{0, 1\}. \tag{16}$$

Since the UMP is an NP-hard problem, $\boldsymbol{P_1}$ will increase exponentially with the increasing number of EVs. In the next section, an approximate algorithm is provided to solve the UMP.

### 5.2 EV Assignment Algorithm (EAA)

In $\boldsymbol{P_1}$, we relax the binary linear constraint and require only $x_{ij} \geq 0$ for all $i, j$ [9]. The linear relaxation for Constraint (16) is described as follows:

$$0 \leq x_{ij} \leq 1, \quad i \in E, \ j \in F. \tag{17}$$

Then, we have the relaxed $\boldsymbol{P_1}$ as follows:

$$\boldsymbol{P_2}: \min_{x_{ij}} \quad T_{max} \tag{18}$$

$$\text{s.t.} \quad (13), (14), (15), (17). \tag{19}$$

$\boldsymbol{P_2}$ can be solved to optimality by CVX. The solution of $\boldsymbol{P_2}$ determines the assignments of all EVs and their corresponding plugs. The outcome of $\boldsymbol{P_2}$ is a set of continuous values. Now, we only have to approximate a set of continuous solutions as a set of integer solutions [8,10]. The rounding technique has three parts:

1) *Continuous Values:* From $\boldsymbol{P_2}$, we can obtain an $m \times n$ matrix $X = (x_{ij})$, where $x_{ij}$ means $e_i$ is assigned to plug $j$ and $0 \leq x_{ij} \leq 1$.
2) *Form a Bipartite Graph:* We will apply the rounding technique to convert the continuous solutions to integers [8]. A bipartite graph $G = (V, W, E)$ represents the relationships between EVs and plugs. $W = \{w_i : i = 1, 2, ..., n\}$ is one side of the bipartite graph, representing the EV nodes. $V = \{v_{js} : j = 1, 2, ..., m; s = 1, ..., k_j\}$ is the other side of the bipartite graph, where $k_j = \lceil \sum_i^n x_{ij} \rceil$ are virtual nodes associated with plug nodes $j$; that is, $k_i$

nodes $\{v_{js} : s = 1, ..., k_i\}$ are derived from plug node $j, j = 1, ..., m$. Next, we consider the edge values of the nodes. We use $e_{js,i}$ to denote the value between node $w_i$ and node $v_{js}$.

Before constructing, we sort the tasks $p_{ij}$ on each plug in 6nonincreasing order. If $\sum_{i=1}^{n} x_{ij} \leq 1$, there is only one node $v_{j1} \in V$, which means only one EV is assigned to plug $j$. In this case, for each $x_{ij} > 0$, we add an edge between nodes $w_i$ and $v_{j1}$ and set $x^*_{j1,i} = x_{ij}$. Otherwise, for each $s \in \{1, 2, 3..., k_j - 1\}$, we need to find the minimum index $i_s$ that satisfies $\sum_{i=1}^{i_s} \geq s$. For each $i \in \{i_{s-1} + 1, ..., i_s - 1\}$ and $x_{ij} > 0$, we let $E$ contain edges $(v_{js}, w_i)$ and set $x^*_{js,i} = x_{ij}$. Then, we add edge $x^*_{js,i}$ to $E$ if $i = i_s$ and set $x^*_{js,i} = s - \sum_{i=1}^{i_s-1} x_{ij}$. If $\sum_{i=1}^{i_s} x_{ij} > s$, we add one more edge $(w_i, v_{j(s+1)})$ and set $x^*_{j(s+1),i} = \sum_{i=1}^{i_s} x_{ij} - s$. Algorithm 1 presents the construction of a bipartite graph.

3) *Find the Maximum Matching:* After building the bipartite graph, we use the Hungarian algorithm to obtain the maximum matching of the bipartite graph. Since nodes $V$ need more than nodes $W$ when constructing a bipartite graph, all $W = \{w_i : i = 1, 2, ..., n\}$ nodes can be matched. If $(w_i, v_{js}, x^*_{ijs})$ in the matching, we set $x_{ij} = 1$; otherwise, $x_{ij} = 0$. The obtained integer solution $x_{ij}$ indicates that $e_i$ is assigned to plug $j$ if $x_{ij} = 1$ and otherwise is not. Thus, we obtain the assignment schedule.

**Algorithm 1.** Construction of the Bipartite Graph

Set $E = \emptyset$
**if** $\sum_{i=1}^{n} x_{ij} \leq 1$ **then**
  There is only one node $v_{j1}$ corresponding to plug $j$.
  **for** each $x_{ij} > 0$ **do**
    Add edge $(w_i, v_{1j})$ to $E$ and set $x^*_{j1,i} = x_{ij}$.
  **end for**
**else**
  **for** $s \in \{1, 2, ..., k_j - 1\}$ **do**
    Find the minimum index $i_s$ that satisfies $\sum_{i=1}^{i_s} \geq s$
    **if** $s = 1$ **then**
      $i_{s-1} = 0$
    **end if**
    **for** $i \in \{i_{s-1} + 1, ..., i_s - 1\}$ and $x_{ij} > 0$ **do**
      $E$ contains edges $(v_{js}, w_i)$ and $x^*_{js,i} = s - \sum_{i=1}^{i_s-1} s_{ij}$.
    **end for**
    **if** $i = i_s$ **then**
      Add edge $(w_i, v_{js})$ to $E$ and set $x^*_{js,i} = s - \sum_{i=1}^{i_s-1} x_{ij}$.
    **end if**
    **if** $\sum_{i=1}^{i_s} x_{ij} > s$ **then**
      Add one more edge $(w_i, v_{j(s+1)})$ and set $s^*_{j(s+1),i} = \sum_{i=1}^{i_s} s_{ij} - s$.
    **end if**
  **end for**
**end if**

### 5.3 MCV Deployment and EV Assignment

After completing the problem decomposition and solving the plug allocation problem, we need to integrate these two problems into a whole. The above approximation algorithm solves the problem $\boldsymbol{P_1}$, so in this section, we solve the problem $\boldsymbol{P_0}$. Algorithm 2 shows the specific solution process.

**Algorithm 2.** MCV deployment and EV assignment

**Input:** $N$, $F$, $L$
**Output:** Minimum completion time
1: **for** Each $l \in L$ **do**
2: Problem Decomposition
3: Algorithm 1
4: find the minimum completion time
5: **end for**
6: **return** minimum completion time

## 6 Simulation Results

In this section, first, we evaluate the performance of the EAA by comparing it with the optimal value for 15 EVs. Second, the proposed algorithm is compared with the greedy algorithm and random selection algorithm.

### 6.1 Parameters and Settings

We consider a rectangular area with a length of 40 km. Within this region, EVs are randomly distributed. The MCV is outside the zone and will be dispatched from outside. In our experiments, we use 20 EVs and 50 EVs for testing. We consider the initial power $E_i^{init}$ of the EV to be 5 to 30 kWh. To consider different EV models, we use EVs with battery capacities $E_{max}$ ranging from 30 to 80 kWh. We assume that the MCV has enough power to complete all charging tasks and move toward the selected parking lot with a speed $v$ of 60 km/h and that the MCV is equipped with five plugs, each with a charging rate $\gamma$ of 150 kWh/h (2.5 kWh/min). In addition, the EV consumption rate $\alpha$ is 0.2 kWh/km while moving to the parking lot.

### 6.2 Results and Discussion

First, we use 15 EVs to compare the results with the optimal value. In Fig. 2, we can see that the EAA has the smallest gap with the optimal algorithm, comparing the results of the greedy algorithm and the random selection algorithm. In addition, the results of the EAA are gentler; that is, the time difference between the plugs is small.

When the number of EVs increases to 20, we cannot obtain the optimal results, so we compare the EAA, greedy algorithm and random selection algorithm under different numbers of EVs. Under the greedy algorithm and the random selection algorithm, the overall charging time of each plug is calculated. Figure 3 shows the charging time of each plug. The results show that the charging times under the EAA are $(76, 64, 81, 62, 67)$. The results of the greedy algorithm and the random algorithm are $(40, 58, 73, 81, 97)$ and $(28, 131, 80, 73, 38)$, respectively. It can be seen from the results that the maximum charging time of the EAA is 81 min. Compared with 97 min for the greedy algorithm and 131 min for the random algorithm, the EAA has obvious advantages in completing the charging tasks.

Then, we used 50 EVs to verify the superiority of our algorithm, and in the following experiments, we used 50 EVs as a standard for comparison. In Fig. 4, we can see that when the number of EVs increases from 20 to 50, under the EAA, the charging times of the plugs are $(197, 165, 198, 179, 168)$. The results of the greedy algorithm and the random selection algorithm are $(86, 158, 190, 214, 259)$ and $(236, 108, 217, 132, 215)$. The maximum completion time of the EAA is 198 min, while the maximum completion times of the greedy algorithm and the random selection algorithm are 259 min and 236 min, respectively. Although the number of EVs is increased, the performance of the proposed algorithm is still relatively efficient. To avoid accidental outcomes caused by a single experiment, we conducted ten experiments with the same specification, and each time, the distribution of the EVs was different. Then, we averaged the ten results, as shown in Fig. 5.

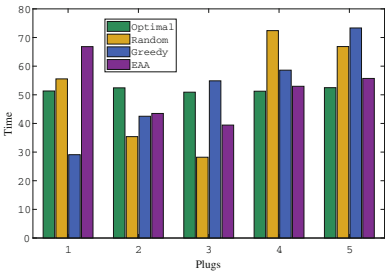



**Fig. 2.** Charging time of each plug when the number of EVs is 15 under the optimal algorithm, greedy algorithm, random selection algorithm and EAA.

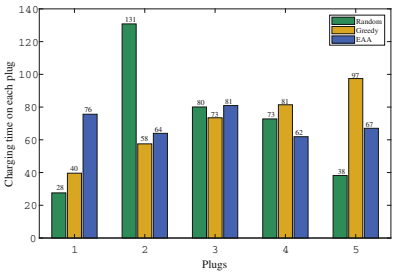



**Fig. 3.** Charging time of each plug when the number of EVs is 20 under the greedy algorithm, random selection algorithm and EAA.

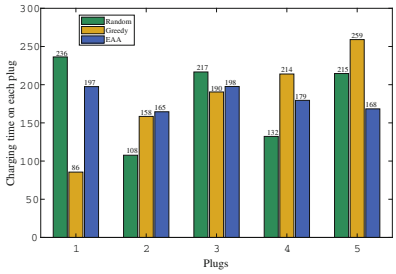



**Fig. 4.** Charging time of each plug when the number of EVs is 50 under the greedy algorithm, random selection algorithm and EAA.

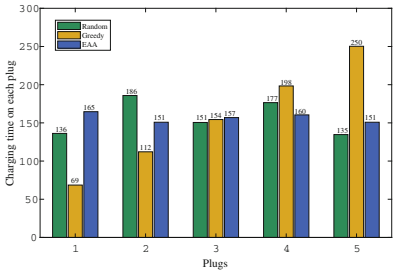



**Fig. 5.** Average charging time in ten experiments for each plug under the random algorithm, greedy algorithm and EAA.

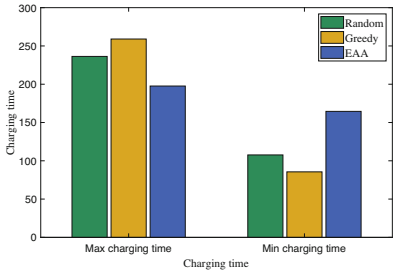



**Fig. 6.** Maximum charging time and minimum charging time under the random algorithm, greedy algorithm and EAA.

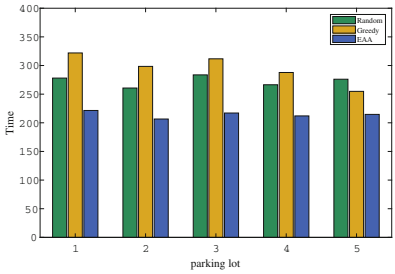



**Fig. 7.** Comparison of the overall time between the EAA, greedy algorithm, and random selection algorithm under different parking lots.

Since our goal is to minimize the maximum charging time, we can compare the differences between the maximum and minimum values of the plugs from the perspective of different algorithms. From Fig. 4, we can see that the differences between the EAA, greedy algorithm, and random selection algorithm are 33 min, 173 min, and 128 min, respectively. According to this result, we can conclude that the time differences between the plugs for the proposed algorithm are relatively small, so the charging time is not distributed unevenly among the plugs, which would cause some plugs to be in an idle state. Figure 6 shows the relative fairness of the proposed algorithm by comparing $T_{max}$ and $T_{min}$ for the three algorithms.

The above comparison is only for problem $\boldsymbol{P_1}$. Next, we will show the efficiency of our algorithm for different parking lots. After completing $\boldsymbol{P_1}$, the movement time of the MCV needs to be integrated; that is, problem $\boldsymbol{P_0}$ is examined.

We randomly generate 5 parking lots in the area and then calculate the time $t$ taken for the MCV to arrive at each parking lot. Then, $T_{max}$ is calculated according to Algorithm 1. We compare the overall time (deployment time $t$ and maximum charging time $T_{max}$) of different algorithms under different parking lots. Figure 7 shows the overall time comparison between the different algorithms.

In the EAA, the overall times for different parking lots are $(221, 206, 217, 212, 214)$. The result of the greedy algorithm is $(321, 298, 311, 287, 254)$. $(278, 260, 283, 266, 276)$ is the result of the random selection algorithm. According to the results, we can clearly see that the EAA will take less time than the greedy algorithm and the random selection algorithm regardless of the selected parking lot. Moreover, the proposed algorithm does not have a very large time difference between different parking lots and is more stable than the greedy algorithm and the random selection algorithm.

## 7 Conclusions

In this paper, we study the deployment of an MCV and the charging scheduling problem in the IoEV. To improve the charging efficiency of the MCV, a one-to-many charging model is proposed. We formulate the deployment problem and the charging schedule as an integer linear programming problem. Since this is an NP-hard problem, an approximate algorithm is applied. We verify the efficiency of our model through the comparison of different characteristics. First, we compare it with the optimal value under 15 EVs. The results show that there is little difference between our results and the optimal value. Then, we use 20 EVs and 50 EVs for comparison. When the amount of data increases, the EAA still performs well. To avoid accidental events, we use ten different groups of data and average the results to verify the advantages of our algorithm. After that, we verify the fairness of our algorithm through the difference between the maximum and minimum times of different plugs. Finally, we consider the impact of different parking lots on the overall goal. Even for different parking lots, our algorithm can still perform well.

## References

1. Huang, S., He, L., Gu, Y., Wood, K., Benjaafar, S.: Design of a mobile charging service for electric vehicles in an urban environment. IEEE Trans. Intell. Transp. Syst. **16**(2), 787–798 (2015). https://doi.org/10.1109/TITS.2014.2341695
2. Liu, L., Qi, X., Xi, Z., Wu, J., Xu, J.: Charging-expense minimization through assignment rescheduling of movable charging stations in electric vehicle networks. IEEE Trans. Intell. Transp. Syst., 1–12 (2022). https://doi.org/10.1109/TITS.2022.3154444
3. Chen, H., Su, Z., Hui, Y., Hui, H.: Dynamic charging optimization for mobile charging stations in internet of things. IEEE Access **6**, 53509–53520 (2018). https://doi.org/10.1109/ACCESS.2018.2868937

4. Zhang, H., et al.: Optimized scheduling for urban-scale mobile charging vehicle. In: 2019 2nd World Symposium on Communication Engineering (WSCE), pp. 164–172 (2019). https://doi.org/10.1109/WSCE49000.2019.9040972
5. Wang, H., Wang, R., Xu, H., Kun, Z., Yi, C., Niyato, D.: Multi-objective mobile charging scheduling on the internet of electric vehicles: a DRL approach. In: 2021 IEEE Global Communications Conference (GLOBECOM), pp. 01–06 (2021). https://doi.org/10.1109/GLOBECOM46510.2021.9685354
6. Ma, Y., Liang, W., Xu, W.: Charging utility maximization in wireless rechargeable sensor networks by charging multiple sensors simultaneously. IEEE/ACM Trans. Netw. **26**(4), 1591–1604 (2018). https://doi.org/10.1109/TNET.2018.2841420
7. Xu, W., Liang, W., Jia, X., Kan, H., Xu, Y., Zhang, X.: Minimizing the maximum charging delay of multiple mobile chargers under the multi-node energy charging scheme. IEEE Trans. Mob. Comput. **20**(5), 1846–1861 (2021). https://doi.org/10.1109/TMC.2020.2973979
8. Shmoys, D.B.: éva Tardos: an approximation algorithm for the generalized assignment problem. Math. Program. **62**(1–3), 461–474 (1993)
9. Pei, Z., Wan, M., Jiang, Z.Z., Wang, Z., Dai, X.: An approximation algorithm for unrelated parallel machine scheduling under TOU electricity tariffs. IEEE Trans. Autom. Sci. Eng. **18**(2), 743–756 (2021). https://doi.org/10.1109/TASE.2020.2995078
10. Dai, Y., Zhang, K., Maharjan, S., Zhang, Y.: Edge intelligence for energy-efficient computation offloading and resource allocation in 5G beyond. IEEE Trans. Veh. Technol. **69**(10), 12175–12186 (2020). https://doi.org/10.1109/TVT.2020.3013990

# Energy-Efficient Algorithms, Systems and Protocol Design

# Energy Efficiency Optimization for RIS Assisted RSMA System over Estimated Channel

Caina Gao, Jia Zhang(✉), Linlin Guo, Lili Meng, Hui Ji, and Jiande Sun

School of Information Science and Engineering, Shandong Normal University, Jinan, Shandong, China
{zhangjia,hui.ji}@sdnu.edu.cn

**Abstract.** In this paper, we consider a reconfigurable intelligent surface (RIS) assisted rate splitting multiple access (RSMA) transmission system with estimated channel state information (CSI). The RIS is used to artificially construct the transmission environment to achieve more energy efficient transmission. An energy efficiency maximization problem is formulated by satisfying the constraint of power budget, the design principles of RSMA and RIS. To solve this problem, fractional programming is first used to decouple the single ratio objective function. Then the optimal power allocation coefficients and the phase shift matrix of RIS are obtained by the proposed alternative optimization method, respectively. Numerical results demonstrate that the energy efficiency performance of the RIS assisted RSMA system can be significantly improved by the proposed alternative joint optimization.

**Keywords:** Energy efficiency · Reconfigurable intelligent surface · Rate splitting multiple access · Imperfect channel state information

## 1 Introduction

Recent studies emphasize the importance of integrating rate splitting multiple access (RSMA) with the next generation wireless technologies, e.g., reconfigurable intelligent surface (RIS) [1–3]. RSMA is a candidate of green potential for the next generation radio access [4,5]. RSMA splits user messages into two parts, common and private, which are encoded respectively at the transmitter side. The common message is encoded into a common data stream and the private messages are encoded into the respective private data streams. The main advantage of RSMA technology is the flexibility of interference management, which can effectively facilitate the system spectral efficiency (SE) and energy efficiency (EE) [6].

This work was supported in part by the NSF of Shandong Province under Grant ZR2021LZH010, Grant ZR2020LZH015, and Grant ZR2020MF042; and in part by the NSF of China under Grant U1736122 and Grant 62071005.


L. Wang et al. (Eds.): WASA 2022, LNCS 13471, pp. 643–654, 2022.
https://doi.org/10.1007/978-3-031-19208-1_53

Energy efficiency is one of the key design metrics for next-generation wireless systems. [7] investigated the EE problem in Cloud Radio Access Networks (C-RAN) under incomplete channel state information (CSI). [8] considered the EE maximization problem in a relay-assisted network, which proposed an energy efficient resource allocation scheme under the power limitation at the transmitter side and the system throughput demand. [9] investigated the energy efficiency problem under three strategies, RSMA, space division multiple access (SDMA) and non-orthogonal multiple access (NOMA), in the multiple-input single-output (MIMO) broadcast channel scenario. It was demonstrated that in a wide range of user deployments, RSMA achieved better energy efficiency compared to SDMA and NOMA.

Reconfigurable intelligent surface [10,11] has been adopted as a promising technology for the sixth generation (6G) wireless communication system. RIS is a device consisting of multiple reflective elements with reconfigurable electromagnetic characteristics that can control the propagation direction of the incident wave. By independently controlling the reflection phase and amplitude of the RIS unit, the radiation pattern can be manipulated. This new technology is able to reduce power consumption and thus improve the system energy efficiency performance. [10] investigated the energy efficiency problem in RIS assisted broadcast communication systems, where the experimental results showed that RIS can significantly improve energy efficiency. [12] considered the energy efficiency maximization problem of RIS assisted multi-cast communication. [13] proposed to design the phase shift and co-variance matrices to investigate the EE-SE trade-off problem in RIS assisted uplink systems. As a passive device, the unknown CSI has become a key challenge for the RIS-assisted system design. Researches on channel estimate in RIS-assisted systems have drawn a lot of attention [14–16].

In this paper, we consider an RIS assisted RSMA transmission system based on the estimated CSI. In the considered RSMA scenario, the common and private messages are transmitted to the users simultaneously, in which the common message is addressed to all users and each private message is transmitted to a single user. Moreover, the RIS is introduced to artificially improve the wireless transmission environment. We formulate an EE maximization problem by jointly optimizing the power allocation among the common and private transmission and the RIS phase shift matrix, which is non-convex. To make the problem tractable, we first use fractional programming to transform the objective function into a decoupled form. Then, we obtain the optimal power allocation coefficients and the phase shift matrix of the RIS by the proposed alternative optimization (AO) based solution. The numerical results demonstrate that the EE performance of the RIS assisted RSMA system can be significantly improved compared to those without joint optimization schemes.

The rest of this paper is organized as follows. Section 2 introduces the RIS assisted RSMA system and signal model. In Sect. 3, we formulate and solve the joint optimization problem of EE maximization. Numerical results are given in Sect. 4 to prove the effectiveness of the proposed algorithm, and we conclude the paper in Sect. 5.

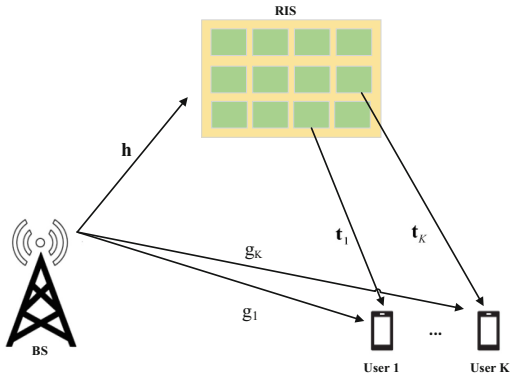



**Fig. 1.** Network model for RIS assisted RSMA transmission.

## 2 RIS Assisted Communication System Model

### 2.1 System Model

As depicted in Fig. 1, we focus on the single-cell downlink transmission scenario. In the considered RSMA based system, the signal is transmitted by the base station (BS) to $K$ single-antenna users. The set of $K$ users in the cell is $\mathcal{K} = \{1, 2, \ldots, K\}$. Moreover, RIS is proposed in the system to assist the wireless transmission with $N$ reflective units. The set of $N$ reflective units is $\mathcal{N} = \{1, 2, \ldots, N\}$.

The message $m_k$ at the BS intended for user $u_k$ is split into two parts including a common part $m_k^c$ and a private part $m_k^p$, and $m_k = m_k^c + m_k^p$. Each private message $m_k^p, k \in \mathcal{K}$, is encoded into the independent private streams $s_1^p, s_2^p, \ldots, s_K^p$, respectively. All common messages $\mathbf{m}^c = [m_1^c, m_2^c, \ldots, m_K^c]$ are jointly encoded into a common stream $s^c$. In this RIS assisted RSMA system, the signal received at $u_k$ is given by

$$y_k = h_k(\sqrt{p^c}s^c + \sum_{k\in\mathcal{K}} \sqrt{p_k^p}\mathbf{s}_k) + n_k, k \in \mathcal{K}, \tag{1}$$

where $n_k \sim \mathcal{CN}(0, \sigma^2)$ is the additive white Gaussian noise (AWGN) with zero mean and variance of $\sigma^2$. $p^c$ and $p_k^p$ are the power allocated to sends the common data stream and each user's private data streams at the BS, respectively. Under the assistance of the RIS, the equivalent channel $h_k$ from the BS to each $u_k$ can be expressed as

$$h_k = \mathbf{t}_k^H \mathbf{\Theta} \mathbf{h} + g_k, k \in \mathcal{K}, \tag{2}$$

where $\mathbf{t_k} \in \mathbb{C}^{N\times 1}$, $\mathbf{h} \in \mathbb{C}^{N\times 1}$, $g_k$ represent the channel responses from RIS to user, BS to RIS and BS to user, respectively. Additionally, $\mathbf{\Theta} = \text{diag}(\theta_1, ..., \theta_N) \in \mathbb{C}^{N\times N}$ is the RIS phase shift matrix, where the phase of the reflection unit $n$ is $\theta_n = e^{j\phi_n}$, and $\phi_n \in [0, 2\pi), n \in \mathcal{N}$.

At the receiver side, each user first decodes the common message while treating the private messages as noise. The signal-to-interference-plus-noise-ratio (SINR) of decoding the common message at $u_k$ is given by

$$\gamma_k^c = \frac{p^c |h_k|^2}{\sum\limits_{j \in \mathcal{K}} p_j^p |h_k|^2 + \sigma^2}, \ k \in \mathcal{K}. \tag{3}$$

The achievable rate of decoding the common message at $u_k$ is

$$R_k^c = \log_2 \left(1 + \gamma_k^c\right), \ k \in \mathcal{K}. \tag{4}$$

To ensure that the common message can be successfully decoded by all users, the achievable data rate of transmitting the common message $R^c$ must satisfy the following requirement

$$R^c \leq \min_{k \in \mathcal{K}} R_k^c, \tag{5}$$

where $R^c = \sum\limits_{k \in \mathcal{K}} r_k^c$, $k \in \mathcal{K}$, and $r_k^c$ indicates the achievable common rate when $u_k$ decodes the common message.

After the common message is decoded, the SINR of decoding the private messages using successive interference cancellation (SIC) at $u_k$ is

$$\gamma_k^p = \frac{p_k^p |h_k|^2}{\sum\limits_{j \in \mathcal{K} \setminus k} p_j^p |h_k|^2 + \sigma^2}, \ k \in \mathcal{K}. \tag{6}$$

The achievable rate of decoding the private message at $u_k$ is

$$r_k^p = \log_2(1 + \gamma_k^p), \ k \in \mathcal{K}. \tag{7}$$

Finally, the overall achievable data rate of user $u_k$ can be derived as

$$R_k = r_k^c + r_k^p, \ k \in \mathcal{K}. \tag{8}$$

### 2.2 Channel Estimation

In this work, we consider the case where the position of the RIS and BS are fixed [17]. RIS can be placed in an open area so that the channel between RIS and BS is in relatively stable condition. Hence, there is no need to estimate $\mathbf{h}$ frequently. In contrast, the wireless channels connecting the RIS and the users need to be estimated more frequently since the mobility of the users. In this context, it is more important to estimate $\mathbf{t}_k$ and $g_k$, respectively.

Given $\mathbf{h}$, the received signal after the $l$th adjustment of the RIS parameters can be expressed as

$$y_k^l = \left(\mathbf{t}_k^H \mathbf{\Theta}_l \mathbf{h} + g_k\right) s + n_k^l, \tag{9}$$

where $s = (\sqrt{p^c} s^c + \sum_{k \in \mathcal{K}} \sqrt{p_k^p} s_k)$, and $\mathbf{\Theta}_l = \text{diag}(\theta_1^l, ..., \theta_N^l)$ for $l = 0, 1, ..., L-1$.

We resort to the assistance of phase changes at the RIS, which can provide additional channel measurements. Without loss of generality, we select $\mathbf{\Theta}_0$ as the reference. Suppose that $L \geq 2$, then we have

$$y_k^0 = \mathbf{t}_k^H \mathbf{\Theta}_0 \mathbf{h} s + g_k s + n_k^0, \tag{10}$$

$$y_k^l = \mathbf{t}_k^H \mathbf{\Theta}_l \mathbf{h} s + \mathbf{g}_k s + n_k^l, l \in \{1, ..., L-1\}. \tag{11}$$

By subtracting (10) from (11), we can remove the unknown term $g_k s$ as follows

$$y_k^0 - y_k^l = \mathbf{t}_k^H \left(\mathbf{\Theta}_0 - \mathbf{\Theta}_l\right) \mathbf{h} s + n_k^0 - n_k^l. \tag{12}$$

For $l = 0, 1, ..., L-1$, $L-1$ subtractions can be obtained following (12). We summarize these subtractions as follows

$$Y_{stack} = A \mathbf{t}_k^H \mathbf{x} + N_{stack}, \tag{13}$$

where
$\mathrm{A} = [(\mathbf{\Theta_0} - \mathbf{\Theta}_1)^T \mathbf{h}^T, ..., (\mathbf{\Theta}_0 - \mathbf{\Theta}_{L-1})^T \mathbf{h}^T]^T$,
$Y_{stack} = \left[\left(y_k^0 - y_k^1\right)^T, \left(y_k^0 - y_k^2\right)^T, \cdots, \left(y_k^0 - y_k^{L-1}\right)^T\right]^T$,
$N_{stack} = [n_k^0 - n_k^1, n_k^0 - n_k^2, \cdots, n_k^0 - n_k^{L-1}]^T$.

Then the least square estimation of $\mathbf{t}_k^H$ can be derived by

$$\hat{\mathbf{t}_k^H} = \left(A^H A\right)^{-1} A^H \left(Y_{stack} - N_{stack}\right) s^{-1}. \tag{14}$$

Once the estimated $\mathbf{t}_k^H$ is obtained, we can quickly calculate the estimation of $g_k$ according to (10) or (11).

## 3 RSMA-Based EE Maximization

### 3.1 Energy Efficient Problem Formulation

With the RSMA based signaling, the EE of the RIS assisted system can be expressed as

$$\eta_{EE} = \frac{\sum\limits_{k \in \mathcal{K}} R_k}{\frac{1}{\eta} P + P_{\mathrm{c}}} = \frac{\sum\limits_{k \in \mathcal{K}} \left(r_k^c + r_k^p\right)}{\frac{1}{\eta} P + P_{\mathrm{c}}}, \tag{15}$$

where $P = p^c + \sum\limits_{k=1}^{K} p_k^p$ is the total transmit power of the BS, $\sum\limits_{k \in \mathcal{K}} R_k$ is the sum rate of all users, and $\eta \in (0, 1)$ denotes the efficiency of the power amplifier. The circuit power consumption $P_c$ includes the dynamic power consumption and the static power consumption, which can be modeled by

$$P_{\mathrm{c}} = \tau \sum_{k \in \mathcal{K}} R_k + P_{\mathrm{s}}, \tag{16}$$

where $\tau$ is a constant denoting the dynamic power consumption per unit throughput, and $P_{\mathrm{s}}$ is the static power at the BS in transmission mode.

We formulate an objective to maximize the EE by jointly optimizing the power (i.e., $p^c$, $p_k^p$) at the BS and the phase-shift matrix (i.e., $\mathbf{\Theta}$) at the RIS. The formulated optimization problem is presented as follows

$$q^* = \underset{\{p^c\},\{p_k^p\},\{\mathbf{\Theta}\}}{\text{maximize}} \frac{\sum\limits_{k\in\mathcal{K}} (r_k^c + r_k^p)}{\frac{1}{\eta}P + P_{\mathrm{c}}} \tag{17a}$$

$$\text{s.t.} \sum_{k\in\mathcal{K}} r_k^c \le R^c, \tag{17b}$$

$$p^c + \sum_{k=1}^{K} p_k^p \le P_T, \tag{17c}$$

$$|\theta_n| = 1, n = 1, ..., N. \tag{17d}$$

In problem (17), constraint (17b) ensures that the common message can be successfully decoded by each single user in the system, constraint (17c) ensures the total transmit power limit. The formulated objective (17a) is a non-convex single ratio fractional function. The constraint (17b) is also non-convex. Therefore, the proposed optimization problem (17) is a non-convex problem that cannot be solved directly. In the next subsection, the alternative optimization (AO) based solution is proposed to solve problem (17) in turn to obtain sub-optimal solutions.

### 3.2 Joint Design of Power Allocation and Reflection Phase Shift Matrix

In general, it is difficult to solve problem (17) directly. Instead, by adopting fractional programming [18], we can first rewrite (17) for a fixed $q$ as follows

$$f(q) = \underset{\{p^c\},\{p_k^p\},\{\mathbf{\Theta}\}}{\text{maximize}} \sum_{k\in\mathcal{K}} (r_k^c + r_k^p) - q\left(\frac{1}{\eta}P + P_{\mathrm{c}}\right) \tag{18a}$$

$$\text{s.t.} \sum_{k\in\mathcal{K}} r_k^c \le R^c, \tag{18b}$$

$$p^c + \sum_{k=1}^{K} p_k^p \le P_T, \tag{18c}$$

$$|\theta_n| = 1, n = 1, ..., N. \tag{18d}$$

We denote $\{p^{cq}, p_k^{pq}, \mathbf{\Theta}^q\}$ as the optimal solution for (18) with the fixed $q$. The optimal value $q^*$ of problem (17) is the solution of the equation $f(q) = 0$. If $q = q^*$, problems (17) and (18) have the same optimal solution, which is $\{p^{c*}, p_k^{p*}, \mathbf{\Theta}^*\} = \{p^{cq}, p_k^{pq}, \mathbf{\Theta}^q\}$.

Problem (18) is also non-convex. In general, there are no standard and effective methods to solve such a complex optimization problem. However, if we fix

the phase shift and optimize the power allocation directly, the corresponding sub-problem is a convex problem. Thus, an AO-based algorithm is proposed to obtain a local optimal solution for problem (18). Specifically, we first optimize the power allocation $\{p^c, p_1^p, ..., p_K^p\}$ with given phase shift matrix $\mathbf{\Theta}$.

Given any feasible RIS phase shift matrix $\mathbf{\Theta}$, the power allocation optimization problem can be written as

$$\underset{\{p^c\},\{p_k^p\}}{\text{maximize}} \sum_{k\in\mathcal{K}} (r_k^c + r_k^p) - q\left(\frac{1}{\eta}P + P_{\text{c}}\right) \tag{19a}$$

$$\text{s.t.} \sum_{k\in\mathcal{K}} r_k^c \le R^c, \tag{19b}$$

$$p^c + \sum_{k=1}^{K} p_k^p \le P_T. \tag{19c}$$

Problem (19) is a convex optimization problem. Therefore, standard convex optimization methods can be used to solve the EE optimization problem, such as the interior point method.

Given the achieved optimal power allocation of problem (18), the phase shift matrix optimization problem can be written as

$$\underset{\{\mathbf{\Theta}\}}{\text{maximize}} \sum_{k\in\mathcal{K}} (r_k^c + r_k^p) - q\left(\frac{1}{\eta}P + P_{\text{c}}\right) \tag{20a}$$

$$\text{s.t.} \sum_{k\in\mathcal{K}} r_k^c \le R^c, \tag{20b}$$

$$|\theta_n| = 1, n = 1, ..., N. \tag{20c}$$

Considering the determined power allocation, the monotonic of the logarithmic function, and the phase shift matrix $\mathbf{\Theta}$ of the RIS is a diagonal matrix, we move to seek the optimal diagonal element $\boldsymbol{\theta} = [\theta_1, ..., \theta_N]^H$ for problem (20). Then, problem (20) can be rewritten as

$$\underset{\boldsymbol{\theta}}{\text{maximize}} \sum_{k\in\mathcal{K}} \log(1 + \frac{p_k^p |h_k|^2}{\sum\limits_{j\in\mathcal{K}\setminus k} p_j^p |h_k|^2 + \sigma^2}) + \sum_{k\in\mathcal{K}} r_k^p \tag{21a}$$

$$\text{s.t.} \sum_{k\in\mathcal{K}} r_k^c \le R^c, \tag{21b}$$

$$|\theta_n| = 1, n = 1, ..., N. \tag{21c}$$

To solve (21), $\mathbf{J}_k = \text{diag}(\mathbf{h^H})\mathbf{t_k}$ is introduced. We have $\mathbf{t}_k^H \mathbf{\Theta} \mathbf{h} = \boldsymbol{\theta}^H \mathbf{J}_k$ and $|h_k|^2 = |\mathbf{t}_k^H \mathbf{\Theta} \mathbf{h} + g_k|^2 = |\boldsymbol{\theta}^H \mathbf{J}_k + g_k|^2$.

By introducing $\bar{\boldsymbol{\theta}} = [\boldsymbol{\theta}; 1]$ and $\mathbf{S}_k = \begin{bmatrix} \mathbf{J}_k \mathbf{J}_k^H & \mathbf{J}_k g_k \\ g_k \mathbf{J}_k^H & 0 \end{bmatrix}$, we have

$$\begin{aligned} |\boldsymbol{\theta}^H \mathbf{J}_k + g_k|^2 &= \bar{\boldsymbol{\theta}}^H \mathbf{S}_k \bar{\boldsymbol{\theta}} + g_k^2 = Tr(\mathbf{S}_k \bar{\boldsymbol{\theta}} \bar{\boldsymbol{\theta}}^H) + g_k^2 \\ &= Tr(\mathbf{S}_k \mathbf{E}) + g_k^2, \end{aligned} \tag{22}$$

where $\mathbf{E} = \bar{\boldsymbol{\theta}}\bar{\boldsymbol{\theta}}^H$. Thus, problem (21) is equivalent to

$$\underset{\mathbf{E}}{\text{maximize}} \quad \sum_{k\in\mathcal{K}}(\log(1+\frac{p_k^p(Tr(\mathbf{S}_k\mathbf{E})+g_k^2)}{\sum\limits_{j\in\mathcal{K}\setminus k} p_j^p(Tr(\mathbf{S}_k\mathbf{E})+g_k^2)+\sigma^2})+r_k^c) \tag{23a}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} r_k^c \le \log(1+\frac{p_k^c(Tr(\mathbf{S}_k\mathbf{E})+g_k^2)}{\sum\limits_{j\in\mathcal{K}} p_j^p(Tr(\mathbf{S}_k\mathbf{E})+g_k^2)+\sigma^2}), \tag{23b}$$

$$|\theta_n| = 1, n = 1, ..., N, \tag{23c}$$

where the optimal RIS phase shift matrix can be obtained by deriving the Lagrangian function and solving it using the Karush-Kuhn-Tucker (KKT) optimal condition.

Our proposed algorithm for the proposed AO-based solution is summarized as follows.

**Algorithm 1.** Proposed AO-based Algorithm

**Initialization:** Set feasible $\left\{\boldsymbol{\theta}_{(0)}, p^{c(0)}, {p_k}^{p(0)}, q^{(0)}\right\}$, $i \leftarrow 1$, and the convergence tolerance $\varepsilon > 0$. Update $q^{(1)}$ of (18) using ${p_k}^{p(0)}$ , $p^{c(0)}$ and $\boldsymbol{\theta}_{(0)}$.

1: **While** $|q^{(i)} - q^{(i-1)}| > \varepsilon$ **Do**
2: Given fixed $\boldsymbol{\theta}_{(i-1)}$, solve (19) and obtain $\left\{p^{c(i)}, p_k^{p(i)}\right\}$.
3: Given the achieved $\left\{p^{c(i)}, p_k^{p(i)}\right\}$, solve (23) and obtain $\boldsymbol{\theta}_{(i)}$.
4: Compute the maximize value of (17) using $\left\{p^{c(i)}, p_k^{p(i)}\right\}$ and $\boldsymbol{\theta}_{(i)}$ and update $q^{(i+1)}$.
5: Set $i \leftarrow i + 1$.
6: **End While**

## 4 Numerical Results

We provide numerical results to evaluate the performance of our proposed solution against three benchmarks. The channel model parameters are consistent as in the reference [19]. We consider a two users system, where BS, RIS, $u_1$ and $u_2$ are located at $(0, 0), (d_0, d_h)$, $(d_1, 0)$ and $(d_2, 0)$ meters (m), respectively. The horizontal distance between the base station and the RIS is $d_h = 50\,\text{m}$. The tolerance of convergence parameter is set as $\varepsilon = 0.0001$.

Figure 2 shows the impact of increasing the number of RIS units on the energy efficiency for the estimated channel. It can be seen that the RIS-assisted system has a significant improvement on the energy efficiency performance compared to the communication system without RIS. Moreover, the EE of the RIS assisted system is maximized by the proposed joint optimizing over the phase shift matrix of the RIS and power allocation coefficients at the BS compared to the ones without joint optimizing. It can also be observed that more reflective units at the

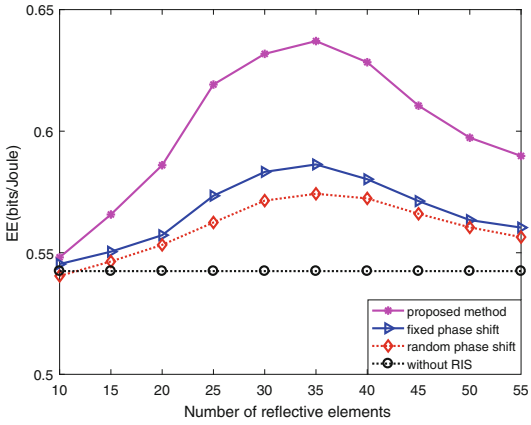



**Fig. 2.** Energy efficiency versus the number of RIS reflective elements $N$, $P_T = 30\,\text{dBm}$, $d_0 = 10$ m, $d_1 = 200\,\text{m}$, $d_2 = 201\,\text{m}$.

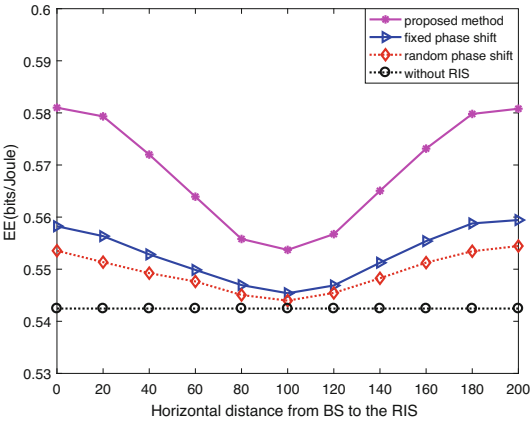



**Fig. 3.** Energy efficiency versus the horizontal distance between BS to RIS with $N = 20$, $P_T = 30\,\text{dBm}$, $d_1 = 200\,\text{m}$, $d_2 = 201\,\text{m}$.

RIS does not necessarily mean a better EE. This is because more reflective units at RIS can provide more degrees of freedom to increase the system capacity, yet too many reflective elements can also increase the total power to ensure quality of service of the communication system, which may reduce the energy efficiency of the RIS assisted system. Thus, a proper number of N needs to be optimized in future work with an objective of EE maximization.

In Fig. 3, it can be found that the proposed algorithm can achieve better energy efficiency when the RIS is close to the base station or the user end. The energy efficiency drops to its minimum when the position of the RIS is in the middle between the BS and the user. In addition, the proposed AO-based solution achieves much better EE performance than the benchmarks.

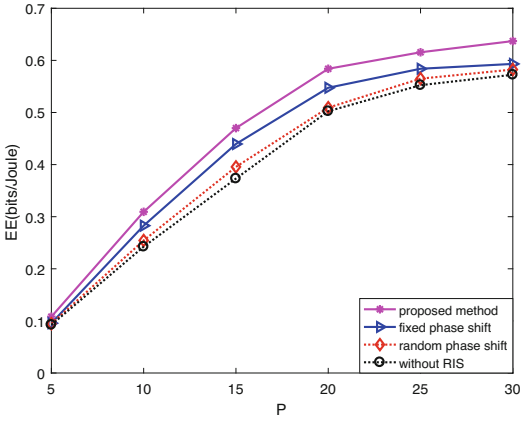



**Fig. 4.** Energy efficiency versus the transmit power $P_T$ with $N = 20$, $d_0 = 10$ m, $d_1 = 200$ m, $d_2 = 201$ m.

Figure 4 investigates the impact on the energy efficiency performance against the maximum transmit power at the BS under the estimated channel state information. In this figure, the energy efficiency increases with increasing transmit power for all schemes. Increasing the transmit power can improve the energy efficiency of the system since more power can provide better system throughput. By joint considering of the numerical results in Fig. 2, we can infer that the performance gap between our proposed algorithm and the benchmarks will get larger when more RIS reflective elements is available.

## 5 Conclusion

In this paper, we consider the EE maximization problem in RSMA system with the assistance of the RIS. The energy efficiency maximization problem is formulated and solved using fractional programming and alternative optimization. Compared to those without joint optimization between power allocation and the RIS design, numerical results show that the energy efficiency performance of the proposed solution can be significantly improved.

## References

1. Zhou, G., Pan, C., Ren, H., Wang, K., Nallanathan, A.: Intelligent reflecting surface aided multigroup multicast MISO communication systems. IEEE Trans. Signal Process. **68**, 3236–3251 (2020)
2. Pan, C., Ren, H., Wang, K., Xu, W., Hanzo, L.: Multicell MIMO communications relying on intelligent reflecting surfaces. IEEE Trans. Wireless Commun. **19**(8), 5218–5233 (2020)

3. Huang, C., Zappone, A., Alexandropoulos, G.C., Debbah, M., Yuen, C.: Reconfigurable intelligent surfaces for energy efficiency in wireless communication. IEEE Trans. Wireless Commun. **18**(8), 4157–4170 (2019)
4. Joudeh, H., Clerckx, B.: Rate-splitting for max-min fair multigroup multicast beamforming in overloaded systems. IEEE Trans. Wireless Commun. **16**(11), 7276–7289 (2017)
5. Mao, Y., Clerckx, B., Li, V.O.K.: Rate-splitting multiple access for downlink communication systems: bridging, generalizing, and outperforming SDMA and NOMA. EURASIP J. Wirel. Commun. Network. **2018**, 133 (2018)
6. Chang, Z., Ristaniemi, T.: Energy efficiency of unicast support multicast with QoS guarantee. In: 2013 IEEE/CIC International Conference on Communications in China - Workshops (CIC/ICCC), Xi'an, China, pp. 16–20 (2013). https://doi.org/10.1109/ICCChinaW.2013.6670559
7. Al-Oquibi, B., Amin, O., Dahrouj, H., Al-Naffouri, T.Y., Alouini, M.: Energy efficiency for cloud-radio access networks with imperfect channel state information. In: 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Valencia, Spain, pp. 1–5 (2016). https://doi.org/10.1109/PIMRC.2016.7794612
8. Zhao, M., Zhao, J., Zhou, W., Zhu, J., Zhang, S.: Energy efficiency optimization in relay-assisted networks with energy harvesting relay constraints. China Commun. **12**(2), 84–94 (2015)
9. Mao, Y., Clerckx, B., Li, V.O.K.: Energy efficiency of rate-splitting multiple access, and performance benefits over SDMA and NOMA. In: 2018 15th International Symposium on Wireless Communication Systems (ISWCS), Lisbon, Portugal, pp. 1–5 (2018). https://doi.org/10.1109/ISWCS.2018.8491100
10. Du, L., Huang, C., Guo, W., Ma, J., Ma, X.: Reconfigurable intelligent surfaces assisted secure multicast communications. IEEE Wirel. Commun. Lett. **9**(10), 1673–1676 (2020)
11. Shen, H., Xu, W., Gong, S., He, Z., Zhao, C.: Secrecy rate maximization for intelligent reflecting surface assisted multi-antenna communications. IEEE Commun. Lett. **23**(9), 1488–1492 (2019)
12. Wang, Y., Lu, H., Zhao, D., Sun, H.: Energy efficiency optimization in IRS-enhanced mmWave systems with lens antenna array. In: 2020 IEEE Global Communications Conference (GLOBECOM), Taipei, Taiwan, pp. 1–6 (2020). https://doi.org/10.1109/GLOBECOM42002.2020.9348266
13. You, L., Xiong, J., Ng, D.W.K., Yuen, C., Wang, W., Gao, X.: Energy efficiency and spectral efficiency tradeoff in RIS-aided multiuser MIMO uplink transmission. IEEE Trans. Signal Process. **69**, 1407–1421 (2021)
14. Jin, Y., Zhang, J., Huang, C., Yang, L., Xiao, H., Ai, B.: Multiple residual dense networks for reconfigurable intelligent surfaces cascaded channel estimation. IEEE Trans. Veh. Technol. **71**(2), 2134–2139 (2022)
15. Shao, X., Cheng, L., Chen, X., Huang, C., Kwan Ng, D.W.: A Bayesian tensor approach to enable RIS for 6G massive unsourced random access. In: 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, pp. 1–7 (2021). https://doi.org/10.1109/GLOBECOM46510.2021.9685371
16. Zhang, J., Qi, C., Li, P., Lu, P.: Channel estimation for reconfigurable intelligent surface aided massive MIMO system. In: 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Atlanta, GA, USA, pp. 1–5 (2020). https://doi.org/10.1109/SPAWC48557.2020.9154276

17. Jin, Y., Zhang, J., Zhang, X., Xiao, H., Ai, B., Ng, D.W.K.: Channel estimation for semi-passive reconfigurable intelligent surfaces with enhanced deep residual networks. IEEE Trans. Veh. Technol. **70**(10), 11083–11088 (2021)
18. Shen, K., Yu, W.: Fractional programming for communication systems-Part I: power control and beamforming. IEEE Trans. Signal Process. **66**(10), 2616–2630 (2018)
19. Du, L., Zhang, W., Ma, J., Tang, Y.: Reconfigurable intelligent surfaces for energy efficiency in multicast transmissions. IEEE Trans. Veh. Technol. **70**(6), 6266–6271 (2021)

# Author Index