

Bazhong Shen
Baocang Wang
Jinguang Han
Yong Yu (Eds.)

Communications in Computer and Information Science

1105

Frontiers in Cyber Security

Second International Conference, FCS 2019
Xi'an, China, November 15–17, 2019
Proceedings



Springer

Communications in Computer and Information Science

1105

Commenced Publication in 2007

Founding and Former Series Editors:

Phoebe Chen, Alfredo Cuzzocrea, Xiaoyong Du, Orhun Kara, Ting Liu,
Krishna M. Sivalingam, Dominik Ślęzak, Takashi Washio, Xiaokang Yang,
and Junsong Yuan

Editorial Board Members

Simone Diniz Junqueira Barbosa 

*Pontifical Catholic University of Rio de Janeiro (PUC-Rio),
Rio de Janeiro, Brazil*

Joaquim Filipe 

Polytechnic Institute of Setúbal, Setúbal, Portugal

Ashish Ghosh

Indian Statistical Institute, Kolkata, India

Igor Kotenko 

*St. Petersburg Institute for Informatics and Automation of the Russian
Academy of Sciences, St. Petersburg, Russia*

Lizhu Zhou

Tsinghua University, Beijing, China

More information about this series at <http://www.springer.com/series/7899>

Bazhong Shen · Baocang Wang ·
Jinguang Han · Yong Yu (Eds.)


Frontiers in Cyber Security

Second International Conference, FCS 2019
Xi'an, China, November 15–17, 2019
Proceedings

Editors

Bazhong Shen
Xidian University
Xi'an, China

Jinguang Han 
Queen's University Belfast
Belfast, UK

Baocang Wang 
Xidian University
Xi'an, China

Yong Yu 
Shaanxi Normal University
Xi'an, China

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-981-15-0817-2 ISBN 978-981-15-0818-9 (eBook)
<https://doi.org/10.1007/978-981-15-0818-9>

© Springer Nature Singapore Pte Ltd. 2019

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd. The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

The Second International Conference on Frontiers in Cyber Security (FCS 2019) was held in Xi'an, P.R. China, November 15–17, 2019. The conference was organized by the State Key Laboratory of Integrated Services Networks and Cryptographic Research Center and Xidian University, and supported by the University of Electronic Science and Technology of China, Shannxi Normal University, Xuchang University, Xi'an University of Posts & Telecommunications, and Queen's University Belfast. In view of the cyber security situation, a permanent theme for FCS is “Cyber Security,” aiming to introduce security concepts and technological achievements from the international forefront in the field of information security, as well as provide insight into the latest development trends and innovative technology of cyber security. The FCS conference series provides a good platform for researchers and practitioners to exchange their latest research achievements and discuss these questions of network security, system security, cryptography, their applications, etc.

This year we received 67 submissions and withdrew 5 manuscripts. All the submissions were anonymous and only the Program Committee (PC) chairs knew the authors' information. Each submission was allocated to at least three Program Committee members and each paper received on average 3.55 reviews. The submission and review process was supported by the EasyChair conference management system. In the first phase, the PC members individually evaluated the papers and did not know the review opinions of others. In the second phase, the papers were carefully checked in an extensive discussion. Finally, the PC decided to accept 20 full papers and 2 short papers, leading to an overall acceptance rate of 35.5%.

The program included two keynote speeches, given by Prof. Xiaojiang Du (Temple University, USA) titled “Anomaly Detection for Applied Smart Home IoTs,” and Prof. Yi Qian (University of Nebraska-Lincoln, USA) titled “Data-driven Network Intelligence for Cyber Security.”

We would like to thank the PC members and the external reviewers for their careful reviews and post-review discussions. The review work is very tough and time-consuming. We also want to deeply thank the members of the Organizing Committee for their excellent service and help for the organization of this conference. We are very grateful to the staff at Springer for their help in producing the proceedings. Finally, and most importantly, we want to thank all the authors who submitted to the conference and made the event a success.

November 2019

Bazhong Shen
Baocang Wang
Jinguang Han
Yong Yu

Organization

General Co-chair

Bazhong Shen Xidian University, China

Program Co-chairs

Jinguang Han Queen's University Belfast, UK
Baocang Wang Xidian University, China
Yong Yu Shaanxi Normal University, China

Public Chair

Xu An Wang Engineering University of CAPF, China

Organizing Chairs

Juntao Gao Xidian University, China
Jie Chen Xidian University, China
Lihua Dong Xidian University, China

Program Committee

Zhenfu Cao East China Normal University, China
Jintai Ding University of Cincinnati, USA
Genyuan Du Xuchang University, China
Christian Esposito University of Salerno, Italy
Giuseppe Fenza University of Salerno, Italy
Massimo Ficco University of Campania Luigi Vanvitelli, Italy
Shaojing Fu National University of Defense Technology, China
Fuchun Guo University of Wollongong, Australia
Rui Guo Xi'an University of Posts and Telecommunications,
China
Debiao He Wuhan University, China
Xinyi Huang Fujian Normal University, China
Qiong Huang South China Agricultural University, China
SK Hafizul Islam Indian Institute of Information Technology Kalyani,
India
Muhammad Khurram Khan King Saud University, Saudi Arabia
Rongxing Lu University of New Brunswick, Canada
Ximeng Liu Singapore Management University, Singapore

Yongjian Liao	University of Electronic Science and Technology of China, China
Fagen Li	University of Electronic Science and Technology of China, China
Shujun Li	University of Kent, UK
Mingzhe Liu	Chengdu University of Technology, China
Yi Mu	University of Wollongong, Australia
Jianbing Ni	University of Waterloo, Canada
Nadia Nedjah	State University of Rio de Janeiro, Brazil
Marek Ogiela	AGH University of Science and Technology, Poland
Longjiang Qu	National University of Defense Technology, China
Arun Kumar Sangaiah	Vellore Institute of Technology, India
Chunhua Su	Osaka University, Japan
Willy Susilo	University of Wollongong, Australia
Meiqin Wang	Shandong University, China
Huaqun Wang	Nanjing University of Posts and Telecommunications, China
Liang Xue	University of Waterloo, Canada
Haomiao Yang	University of Electronic Science and Technology of China, China
Dong Zheng	Xi'an University of Posts and Telecommunications, China
Sherali Zeadally	University of Kentucky, USA
Zhili Zhang	Xuchang University, China
Mingwu Zhang	Hubei University of Technology, China
Fanguo Zhang	Sun Yat-sen University, China
Lei Zhang	East China Normal University, China

Additional Reviewers

Yu Chen	Meiyan Xiao
Yuzhao Cui	Shaohao Xie
Tong Chen	Yu Yu
Qingwen Guo	Xu Yang
Jianye Huang	S. J. Yang
Burong Kang	Zhichao Yang
Chao Lin	Yiou Zhao
Shaopeng Liang	Yudi Zhang
Hongbo Li	Zheng Zhang
Jinhua Ma	Zhuoran Zhang
Xinyu Meng	Yan Zhang
Ou Ruan	Yuexin Zhang
Hua Shen	Rui Zhang
Yangtong Tian	Huang Zhang
Shixiong Wang	

Keynote Speech Abstracts

Anomaly Detection for Applied Smart Home IoTs

Xiaojiang Du

Temple University, USA

Abstract. With the large-scale deployment of Internet of Things, smart home has become a popular trend that enables pervasive interactions among home IoT devices. The emergence of home automation platforms brings more benefits of inter-operability among heterogeneous devices and automation programs (also called smart apps). However, as the integration system is tightly coupled with the physical environment, device anomalies occur for varieties of reasons such as device malfunctions and spurious commands and may lead to severe consequences if not handled timely. Prior works utilize data mining approaches to detect problematic device actions and faulty sensor events but suffer from high false alarm rate. Our observation is that data mining based approaches miss a large chunk of information about smart apps and related platform information. In this work, we propose a semantics-aware anomaly detection system for applied home automation platforms that models the home automation system's normal behaviors from both the smart apps source code and history events logs. We evaluate our design with a prototype implementation on Samsung SmartThings platform and test it against 15 anomalous cases of 4 categories. The results show that our system achieves an average accuracy higher than 96% on all 15 anomalous cases while having a very low false alarm rate compared to state-of-art works.

Data-Driven Network Intelligence for Cyber Security

Yi Qian

Department of Electrical and Computer Engineering,
University of Nebraska-Lincoln

Abstract. Data-driven network intelligence will offer a robust, efficient, and effective computing system for anomaly detection in cyber security applications. In this talk, we first summarize the current development and challenges of network intelligence for anomaly detection. Based on the current development, we present a data-driven intelligence system for network anomaly detection. With the support of extended computing, storage, and other resources to the network edge, fog computing is incorporated into the design of the system. The proposed system consists of three types of major components: edge enabled infrastructure, AI engines, and decision platforms. Edge enabled infrastructure provides efficient and effective computing resources for parallel computing and data storage. AI engines produce optimal learning models for threat detection, and enable online machine learning for efficient model update. Decision platforms offer real-time network monitoring, anomaly detection, and threat mitigation. We demonstrate that the envisioned data-driven network intelligence system achieves high detection accuracy and provides robust computational performance for cyber security.

Cyber Security Defense: From Moving Target Defense to Cyber Deception

Jie Wu

Temple University

Abstract. Deception technology is an emerging field of cyber security defense. Products from deception technology can detect, analyze, and defend against zero-day and advanced attacks. The talk starts with the discussion of some unique challenges with cyber deception, as compared with some other deception technology such as in military. We then focus on moving target defense with a couple of examples and some recent results. Finally, we discuss several future directions of cyber deception research with a focus on game and theoretical models.

Bio: Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program cochair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

Contents

Symmetric Key Cryptography

- Improving File Hierarchy Attribute-Based Encryption Scheme
with Multi-authority in Cloud 3
Li Kang and Leyou Zhang

Public Key Cryptography

- Anonymous Leakage-Resilient Ciphertext-Policy Attribute-Based
Encryption Supporting Direct Revocation 21
Xiaoxu Gao, Leyou Zhang, and Gongcheng Hu
- Cryptographic Reverse Firewalls for Identity-Based Encryption 36
Yuyang Zhou, Yuanfeng Guan, Zhiwei Zhang, and Fagen Li
- Identity-Based Encryption Resilient to Continual Leakage Without
Random Oracles 53
Yuyan Guo, Mingming Jiang, Shimin Wei, Ming Xie, and Mei Sun

Post-quantum Cryptography

- CLIBDA: A Deniable Authentication Scheme for Pervasive
Computing Environment 67
Emmanuel Ahene, Yuanfeng Guan, Zhiwei Zhang, and Fagen Li
- Leveled Lattice-Based Linearly Homomorphic Signature Scheme
in the Standard Model for Network Coding 84
Fenghe Wang, Shaoquan Shi, and Chunxiao Wang
- Symmetric Lattice-Based PAKE from Approximate Smooth Projective
Hash Function and Reconciliation Mechanism 95
Zilong Wang, Honggang Hu, Mengce Zheng, and Jiehui Nan
- Zero-Knowledge Proofs for Improved Lattice-Based Group Signature
Scheme with Verifier-Local Revocation 107
Yanhua Zhang, Yifeng Yin, Ximeng Liu, Qikun Zhang, and Huiwen Jia
- Post-Quantum Pseudorandom Functions from Mersenne Primes 128
Jiehui Nan, Mengce Zheng, and Honggang Hu

Signature

An Efficient Proxy Re-Signature Over Lattices 145
Mingming Jiang, Jinqiu Hou, Yuyan Guo, Yan Wang, and Shimin Wei

Batch Verification of Linkable Ring Signature in Smart Grid 161
Qiyu Wang, Jie Chen, and Lishuang Zhuang

Hierarchical Identity-Based Signature over Verifiable Random Function. 177
Juan Ren and Leyou Zhang

Attack and Behavior Detection

Analysis of Ciphertext Policy Hidden Attribute-Based Encryption
and Its Improved Method. 193
Gongcheng Hu and Leyou Zhang

Implementing Attacks on the Approximate Greatest Common
Divisor Problem 209
Leizhang Wang, Quanbo Qu, Tuoyan Li, and Yange Chen

M4D: A Malware Detection Method Using Multimodal Features 228
Yusheng Dai, Hui Li, Xing Rong, Yahong Li, and Min Zheng

New Key Recovery Attack on the MICKEY Family of Stream Ciphers 239
Lin Ding, Dawu Gu, and Lei Wang

Authenticated Key Agreement

A General Construction for Password-Based Authenticated Key
Exchange from Witness PRFs. 253
Jiehui Nan, Mengce Zheng, Zilong Wang, and Honggang Hu

Certificateless Authenticated Key Agreement for Decentralized WBANs 268
Mwitende Gervais, Liang Sun, Ke Wang, and Fagen Li

Blockchain

A Certificateless Proxy Re-encryption Scheme
for Cloud-Based Blockchain. 293
Nabeil Eltayieb, Liang Sun, Ke Wang, and Fagen Li

A Novel Fair and Verifiable Data Trading Scheme 308
Haiyong Yu, Juntao Gao, Tong Wu, and Xuelian Li

Public Audit Scheme of Shared Data Based on Blockchain 327
Junfeng Tian, Xuan Jing, and Ruifang Guo

System and Network Security

A Hybrid Key Management Scheme for Wireless Sensor Network 347
Yanyan Han, Yanru He, Peihe Liu, Xiaoxuan Yan, and Na Li

Author Index 365

Symmetric Key Cryptography



Improving File Hierarchy Attribute-Based Encryption Scheme with Multi-authority in Cloud

Li Kang^(✉) and Leyou Zhang

School of Mathematics and Statistics, Xidian University, Xi'an 710126, China
li_kkang@126.com

Abstract. With the rapid development of cloud computing technology, users tend to store their data remotely in the cloud to save storage space and enjoy scalable services. However, the cloud servers are not entirely trusted. Ciphertext-policy attribute-based encryption (CP-ABE) is considered as an effective cryptographic approach to prevent the untrusted cloud servers from leaking private data. Since in some areas such as medical and business, the shared data has the feature of multi-level hierarchy, so it makes sense to construct a hierarchy ABE scheme. Recently, Guo et al. proposed a PHR hierarchy multi-authority CP-ABE scheme, which implements global identifier (GID) hiding and hierarchical access control. Unfortunately, we find that the recursive operation ($DecryptNode(CT, SK, (x, y))$) defined in their scheme during the decryption phase is doubtful. Based on the analysis, we propose an improving file hierarchy MA-ABE scheme. The scheme preserves the security and privacy of the original scheme but reduces the user's decryption overhead. In addition, we solve the shortcoming which exists in Guo's scheme and the other corresponding schemes.

Keywords: File hierarchy · Attribute-based encryption · Multi-authority · Cloud computing

1 Introduction

A person's identity can be identified by certain attributes. This concept was first introduced by Sahai and Waters [1] in 2005. Since then, the attribute-based encryption (ABE) scheme, as a new public-key encryption system, has been widely used in the cloud storage system as it supports fine-grained access control. In general, according to whether the access structure is related to attributes or to ciphertext, ABE scheme is divided into two types. One is the key-policy ABE (KP-ABE) proposed firstly by Goyal et al. [2] and the other is the ciphertext-policy ABE (CP-ABE) proposed firstly by Bethencourt et al. [3]. In most data

Supported by the National Cryptography Development Fund under grant (MMJJ20180209).

sharing systems, the CP-ABE scheme performs better since the data owner can define the access structure himself/herself to determine the recipients who can successfully access the data.

In a single-authority ABE scheme, the central authority (CA) is responsible for authenticating all users and distributing their private keys. Undoubtedly, this brings an excessive burden and a potential risk of corruption. Moreover, in practice, an encryption system often involves multiple different domains, so the single-authority ABE scheme is no longer applicable. To solve these problems, Chase [4] put forward a multi-authority ABE (MA-ABE) scheme in 2007. In this scheme, multiple authorities replace the single authority responsible for managing attributes and generating private keys for users. But it still needs a CA to generate public-private key pairs for the multiple authorities. In 2009, Chase and Chow [5] introduced a privacy-preserving (PP) MA-ABE scheme to remove CA and hide user GID privacy using a distributed pseudorandom functions (PRF) and 2-party secure computing (2PC) technique, respectively. This is the first scheme that takes user privacy into account. In 2015, Qian et al. [6] constructed a PP-PHR sharing scheme with multi-authority in the same way. Different from the previous schemes, Lewko and Waters [7] proposed a decentralized ABE scheme, in which CA is not needed and no cooperation among multiple authorities. Later, a lot of decentralized KP-ABE schemes [8–11] and decentralized CP-ABE schemes [12–15] dedicated to protecting user privacy were proposed.

Since the shared data files usually have the characteristic of multi-level hierarchy, especially in enterprise and medical domains, it is necessary to construct an ABE scheme that supports file hierarchy. The idea of hierarchical encryption was first introduced by Gentry and Silverberg [16], who constructed a hierarchical identity-based encryption (HIBE) scheme. In 2010, Wang et al. [17] first put forward a hierarchical attribute-based encryption (HABE) scheme by combining the HIBE and CP-ABE schemes to support data sharing on cloud servers. Wan et al. [18] presented a hierarchical attribute-set-based encryption (HASBE) to achieve inherit flexibility, scalability and fine-grained access control. Wang et al. [19] proposed an efficient file hierarchy ABE scheme, which integrated layered access structures into a single one (shown in Fig. 1) and then used the integrated access structure to encrypt hierarchical files. In this way, the scheme reduced the burden of ciphertext storage and the computation cost of encryption. However, there is only one authority here, which is not suitable for the distributed systems.

As people pay more and more attention to privacy protection, some privacy-preserving HABE schemes [20–22] were proposed. In 2016, Zhang et al. [20] combined HIBE and anonymous ABE (AABE) schemes to construct a hierarchical AABE (HAABE) scheme, which has constant-size private keys and short public keys. In 2018, Sandhia et al. [21] proposed a file hierarchy hidden CP-ABE scheme with multi-authority. In this scheme, they defined a novel weighted access structure, where attributes are assigned weights according to their access privileges. The data files are arranged hierarchically according to their attribute weights. Recently, Guo et al. [22] applied the hierarchical ABE scheme to the PHR system. They extended the scheme [19] to a multi-authority system and implemented GID hiding in the same way as the scheme [6].

However, after a deep analysis, we find that the defined recursive operation $DecryptNode(CT, SK, (x, y)) = \prod_{k \in \{1, 2, \dots, N\}} e(C_{(x, y), k}, S_{k, i})$ in the decryption algorithm of the scheme [22] is incorrect as they only focused on the inerrancy of the computational process, but ignored the inherent logic relationship. In the multi-authority ABE scheme, the attributes managed by different authorities are disjoint, and an attribute is monitored by only one attribute authority. Once the attribute node (x, y) is selected, the corresponding k is unique. Therefore, the existence of $\prod_{k \in \{1, 2, \dots, N\}}$ in the formula is unreasonable. The detailed analysis is given in Sect. 4.2. Based on this, an improved file hierarchy attribute-based encryption scheme is proposed.

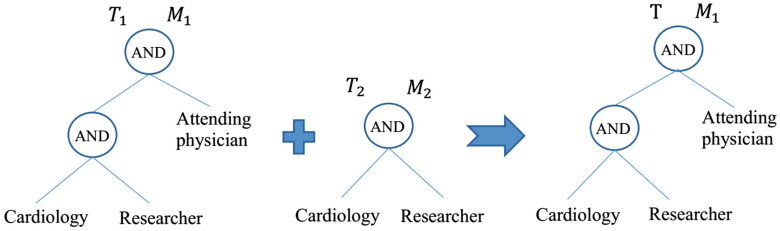


Fig. 1. The integrated access structure

Contributions. By analyzing Guo’s scheme [22], we argue that the scheme has a defect in defining the recursive algorithm $DecryptNode(CT, SK_U, (x, y))$ as the existence of $\prod_{k \in \{1, 2, \dots, N\}}$ in the equation is unreasonable. Then we propose an improving scheme to solve this problem, so that legitimate users can perform correct decryption calculations. In addition, in our construction, before the user runs the decryption algorithm, the cloud server executes a pre-decryption operation, which bears heavy decryption overhead, and then sends the calculation results and ciphertext to the user. The user only needs to perform a simple calculation to get the corresponding plaintext. Therefore, the user’s computation cost is reduced.

2 Preliminaries

2.1 Bilinear Maps

Suppose \mathbb{G} and \mathbb{G}_T are two multiplicative cyclic groups with prime order p . Let g be a generator of group \mathbb{G} and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map, which satisfies the following properties:

- (1) Bilinearity: $\forall g, f \in \mathbb{G}, \forall u, v \in \mathbb{Z}_p$, we have $e(g^u, f^v) = e(g, f)^{uv}$.
- (2) Non-degeneracy: $e(g, g) \neq 1$.
- (3) Symmetry: $e(g^u, f^v) = e(g^v, f^u) = e(g, f)^{uv}$.

Note that for $\forall g, f \in \mathbb{G}$, the operation $e(g, f)$ on group \mathbb{G}_T is efficiently computable.

2.2 Decisional Bilinear Diffie-Hellman (DBDH) Assumption

Suppose g is a generator of group \mathbb{G} , and a, b, c, z are random numbers selected in group \mathbb{Z}_p . The DBDH assumption holds if the advantage of all probabilistic polynomial-time (PPT) algorithm \mathcal{B} distinguish the tuple $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^{abc})$ from the tuple $(A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^z)$ is negligible. Define the advantage of algorithm \mathcal{B} as

$$Adv_{\mathcal{B}}^{DBDH} = |Pr[\mathcal{B}(A, B, C, e(g, g)^{abc}) = 1] - Pr[\mathcal{B}(A, B, C, e(g, g)^z) = 1]|.$$

2.3 Access Structure

Let $\mathbb{P} = \{P_1, P_2, \dots, P_n\}$ denote the set of parties. A collection $\mathbb{A} \subseteq 2^{\mathbb{P}}$ is called monotonic: if $X \in \mathbb{A}$ and $X \subseteq Y$, then $Y \in \mathbb{A}$. The (monotone) access structure is a (monotone) collection \mathbb{A} of non-empty subsets of \mathbb{P} , namely, $\mathbb{A} \subseteq 2^{\mathbb{P}} \setminus \{\emptyset\}$. A set in \mathbb{A} is called the authorized set, and the set that is not in \mathbb{A} is called the unauthorized set.

2.4 Hierarchical Access Tree

Define a hierarchical access tree \mathcal{T} [19], which has l access levels. Each leaf node is described as an attribute, and every non-leaf node represents a threshold gate. We use (x, y) to denote a node of \mathcal{T} . The symbols x and y mark the row and column of node (x, y) in a top-down and left-to-right manner, respectively. Other symbols used in \mathcal{T} are presented in Table 1.

Table 1. Notations.

Symbol	Implication
(x_m, y_m)	The level node of \mathcal{T} ($m \in [1, l]$)
$num_{(x, y)}$	The number of children nodes of node (x, y)
$k_{(x, y)}$	The threshold value of node (x, y) ($0 < k_{(x, y)} \leq num_{(x, y)}$)
$parent(x, y)$	The parent node of (x, y)
Transport node	The node has a child node containing at least one threshold gate
$TNC(x, y)$	A threshold gate set of the children nodes of transport node (x, y)
$att(x, y)$	The attribute value of the leaf node (x, y)
$index(x, y)$	The number associated with node (x, y) ($1 \leq index(x, y) \leq num_{(x, y)}$)
\mathcal{T}_R	An access tree \mathcal{T} rooted at the node R
$\mathcal{T}_{(x, y)}$	The access subtree with (x, y) as the root node

Note that the access levels are arranged in descending order. Namely, (x_1, y_1) occupies the highest level, while (x_l, y_l) occupies the lowest level.

Satisfying a hierarchical access tree. We define $\mathcal{T}_{(x,y)}(S) = 1$ if an attribute set satisfies the access tree $\mathcal{T}_{(x,y)}$. $\mathcal{T}_{(x,y)}(S)$ can be calculated recursively as follows. If (x, y) is a non-leaf node, $\mathcal{T}_{(x,y)}(S) = 1$ when at least $k_{(x,y)}$ children return 1; if (x, y) is a leaf node, $\mathcal{T}_{(x,y)}(S) = 1$ if and only if $att(x, y) \in S$.

3 System Model and Algorithm Definition

3.1 System Model

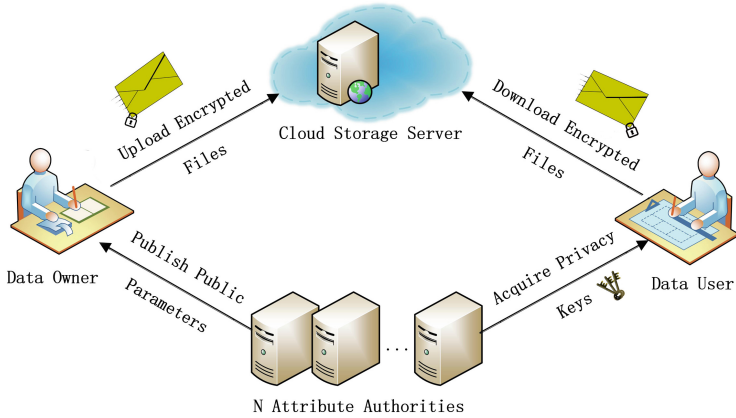


Fig. 2. System model

As shown in Fig. 2, there are 4 entities: Data Owner, N Attribute Authorities, Cloud Storage Service (CSS) and Data User.

1. The data owner defines the access policy and encrypts data file before uploading it to the CSS. In the hierarchy ABE scheme, the owner divides the shared message into l different files and defines the corresponding l access levels according to a reasonable rule. It is natural to assume the data owner is honest.
2. The N authorities manage disjoint attribute sets and are responsible for generating secret keys for users. Similar to the scheme [5], the colluding authorities may aggregate the user's data to "recover" his attribute set by tracking the same GID.
3. The cloud storage server (CSS) is assumed to be an honest-but-curious entity with huge storage space and strong computing power, which provides the service of storing ciphertext for data owners and provides partial decryption service for data users. The CSS works normally except that it tries to gather more ciphertext-related information.

4. The data user can issue secret key queries to the authorities and download any encrypted data files on the CSS. Users can get corresponding data files according to their own access level. In addition, all lower level files are also available to them. In the encryption system, there may exist some dishonest or even malicious users who attempt to collude with others for illegal access.

3.2 Algorithm Definition

The hierarchy ABE (HABE) scheme consists of the following five algorithms: Suppose that there are N authorities $\{A_1, A_2, \dots, A_N\}$ in the system, \tilde{A}_k and \tilde{U} represent the set of attributes owned by the authority A_k ($k = 1, 2, \dots, N$) and the user U , respectively.

Global Setup: This algorithm takes a security parameter λ as input and returns the system parameters PP .

Authority Setup: Each authority A_k runs this algorithm to generate its public-secret key pair (PK_k, SK_k) .

KeyGen: Each authority A_k executes this algorithm with user U to generate the user's secret key. Inputting the system parameters PP , A_k 's secret key SK_k , user's global identifier u and a set of attributes \tilde{U} , this algorithm outputs the secret key SK_U for the user.

Encryption: This algorithm takes as input the system parameters PP , message $M = (M_1, M_2, \dots, M_l)$, A_k 's public keys PK_k and an access structure \mathcal{T} , outputs the ciphertext CT .

Decryption: This algorithm divides into two phases.

- *CSS-Decryption:* This phase is performed by the CSS. On input the system parameters PP , secret key SK_U and the ciphertext CT , if user's attributes satisfy the partial or whole \mathcal{T} , it returns the corresponding decryption results to the user.
- *User-Decryption:* This phase is executed by the user. User takes the ciphertext CT and the results returned by the CSS as input, runs this algorithm and gets the final decryption results.

4 Analysis and an Improving Construction

In this section, we first review the scheme of Guo et al. [22], then give a detailed analysis in Sect. 4.2, and finally put forward an improving scheme.

4.1 Review of Guo's Scheme

Define the Lagrange coefficient $\Delta_{i,S}$ for $i \in \mathbb{Z}_p$ and a set, S , of elements in \mathbb{Z}_p : $\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$. Suppose that there are N authorities in the system and each A_k ($k = 1, 2, \dots, N$) monitors a set of attributes $\tilde{A}_k = (a_{k,1}, \dots, a_{k,n_k})$.

Global Setup. Take a security parameter λ as input, this algorithm returns the public parameters $PP = (e, p, g, h, \mathbb{G}, \mathbb{G}_T)$, where $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a bilinear

map, \mathbb{G} and \mathbb{G}_T are multiplicative cyclic groups with prime order p , g and h are generators of group \mathbb{G} . Let $H_0 : \{0, 1\}^* \rightarrow \mathbb{Z}_p$ and $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_T$ be two strong collision-resistant hash functions. A user with global identity GID has $u = H_0(GID)$.

Authority Setup. Each authority A_k randomly chooses $\alpha_k, t_{k,i} \in_R \mathbb{Z}_p$ and computes $Y_k = e(g, g)^{\alpha_k}, T_{k,i} = g^{t_{k,i}}$, where $t_{k,i}$ is selected for each attribute $a_{k,i} \in \tilde{A}_k$. Each pair of authorities (A_k, A_j) executes a 2-party key exchange protocol to share a secret PRF seed [5] $s_{k,j} (= s_{j,k})$. Authorities A_k and A_j randomly select $x_k, x_j \in \mathbb{Z}_p$ and calculate $y_k = h^{x_k}$ and $y_j = h^{x_j}$ respectively.

Then, they define a pseudorandom function: $PRF_{k,j}(u) = h^{\frac{x_k x_j}{s_{k,j} + u}}$. Finally, A_k publishes the public keys $PK_k = (y_k, Y_k, \{T_{k,i}\}_{i \in 1, 2, \dots, n_k})$ and keeps the master secret keys $SK_k = (\alpha_k, \{t_{k,i}\}_{i \in 1, 2, \dots, n_k}, \{s_{k,j}\}_{i \in \{1, 2, \dots, N\} \setminus \{k\}}, x_k)$.

KeyGen. Let \tilde{U} represent the attribute set of the user. For each attribute $a_{k,i} \in \tilde{A}_U^k (= \tilde{A}_k \cap \tilde{U})$, A_k picks $r_k \in_R \mathbb{Z}_p$ and calculates the attribute secret key $S_{k,i} = h^{\frac{r_k}{t_{k,i}}}$. Then, user runs the anonymous key issuing protocol in [6] with A_k in $N - 1$ times to get the key component: for $k > j$, $D_{kj} = g^{\alpha_k} h^{r_k} PRF_{k,j}(u)$; for $k \leq j$, $D_{kj} = g^{\alpha_k} h^{r_k} / PRF_{k,j}(u)$. Finally, the user computes $D_U = \prod_{(k,j) \in \{1, 2, \dots, N\} \times (\{1, 2, \dots, N\} \setminus \{k\})} D_{k,j} = g^{\sum (N-1)\alpha_k} \cdot h^{\sum (N-1)r_k}$. The user's secret key is $SK_U = (D_U, \{S_{k,i}\}_{k \in [1, N], a_{k,i} \in \tilde{A}_U^k})$.

Encryption. The encryptor first defines a tree access structure \mathcal{T} , under which the data he/she wants to share is encrypted. Suppose that the owner divides data into l files $M = (M_1, M_2, \dots, M_l)$ with l access levels and sets level nodes (x_m, y_m) ($m \in [1, l]$) in \mathcal{T} . For each node (x, y) , owner randomly selects a polynomial $q_{(x,y)}$ of degree $d_{(x,y)} = k_{(x,y)} - 1$, where $k_{(x,y)}$ is the threshold value. For the root node R , owner picks $s_1 \in_R \mathbb{Z}_p$ and sets $q_R(0) = q_{(x_1, y_1)}(0) = s_1$. For the node $(x, y) \in \mathcal{T} \setminus R$, if it is a leaf node, sets $q_{(x,y)}(0) = q_{(x_m, y_m)}(0) = s_m$, otherwise sets $q_{(x,y)}(0) = q_{parent(x,y)}(index(x, y))$. Let $TNC(x, y) = \{child_1, child_2, \dots, child_v, \dots\}$. The encryptor computes

$$C_m^1 = M_m \prod_{k \in \{1, 2, \dots, N\}} Y_k^{s_m},$$

$$C_m^2 = g^{s_m}, C_{(x,y),k} = T_{k,i}^{q_{(x,y)}(0)},$$

$$C_{(x,y),v} = \left(\prod_{k \in \{1, 2, \dots, N\}} Y_k^{q_{(x,y)}(0) + q_{child_v}(0)} \cdot H_1 \left(\left(\prod_{k \in \{1, 2, \dots, N\}} Y_k^{q_{(x,y)}(0)} \right) \right) \right)$$

The ciphertext is $CT = (C_m^1, C_m^2, \{C_{(x,y),k}\}_{a_{k,i} \in \tilde{A}_\mathcal{T}}, C_{(x,y),v})$, where $a_{k,i} = att(x, y)$ is the attribute of the leaf node (x, y) , $\tilde{A}_\mathcal{T}$ is the attribute set in \mathcal{T} .

Decryption. To decrypt the ciphertext, user first defines a recursive algorithm $DecryptNode(CT, SK_U, (x, y))$. For the leaf node (x, y) , if $a_{k,i} \in \tilde{A}_U^k$,

$$DecryptNode(CT, SK_U, (x, y)) = \prod_{k \in \{1, 2, \dots, N\}} e(C_{(x,y),k}, S_{k,i})$$

$$= \prod_{k \in \{1, 2, \dots, N\}} e(T_{k,i}^{q(x,y)(0)}, h_{t_{k,i}}^{r_k}) = e(g, h)^{q(x,y)(0) \cdot \sum r_k}$$

If $a_{k,i} \notin \tilde{A}_U^k$, output $\text{DecryptNode}(CT, SK_U, (x, y)) = \perp$.

For the non-leaf node (x, y) , computes $F_{(x,y)} = \prod_{z \in S_{(x,y)}} F_z^{\Delta_{i,S'_{(x,y)}}(0)} = e(g, h)^{q(x,y)(0) \cdot \sum r_k}$. Continue to call the recursive algorithm, if the subtree is satisfied, $A_m = \text{DecryptNode}(CT, SK_U, (x_m, y_m)) = e(g, h)^{s_m \cdot \sum r_k}$ can be obtained. Then, user calculates $F_m = \frac{e(C_m^2, D_U)^{\frac{1}{N-1}}}{A_m} = e(g, g)^{s_m \cdot \sum \alpha_k}$.

If \tilde{U} includes the lower authorization nodes, user can recursively calculate the values of $F_{(m+1),v}, \dots, F_{(l),v}$, where $F_{(m+1),v} = \frac{C_{(x,y),v}}{F_m \cdot H_1(F_m)} = e(g, g)^{q_{child_v}(0) \cdot \sum \alpha_k}$. That is the values F_m, F_{m+1}, \dots, F_l can be obtained. Finally, the original data file can be restored as $M_m = \frac{C_m^1}{F_m}$.

4.2 A Defect in Their Scheme

In the **Decryption** stage, they defined a recursive algorithm $\text{DecryptNode}(CT, SK, (x, y))$. If (x, y) is a leaf node, then:

$$\text{DecryptNode}(CT, SK, (x, y)) = \prod_{k \in \{1, 2, \dots, N\}} e(C_{(x,y),k}, S_{k,i}) = e(g, h)^{q(x,y)(0) \cdot \sum r_k}$$

However, in most multi-authority systems, N authorities manage disjoint attribute sets, and an attribute can only come from one authority. For instance, in a PHR system, the domains involved include hospital, police station, insurance company, etc. The authorities manage the set of attributes in their field and do not overlap with each other. Take the user's ID number as an example, it is only managed by the police station. Once a leaf node (x, y) (namely, attribute $a_{k,i}$) is selected, the corresponding attribute management authority A_k is unique. There is no ciphertext component in the form of $C_{(x,y),j}$, where $j \in \{1, 2, \dots, N\} \setminus \{k\}$. So, it does not make sense to use $\prod_{k \in \{1, 2, \dots, N\}}$ here.

In fact, the purpose of the author to introduce $\prod_{k \in \{1, 2, \dots, N\}}$ here is to ensure that a common power $\sum_{k \in \{1, 2, \dots, N\}} r_k$ (like the $\sum d_k$ in scheme [23]) can be obtained, so that when the recursive algorithm is called, the user can successfully recover the secret value s_m by using Lagrange interpolation. In the multi-authority system, all values except $q(x,y)(0)$ must be the same when calculating $\text{DecryptNode}(CT, SK, (x, y))$, otherwise the secret value s_m can not be recovered. Next, we will give a simple example to illustrate our points.

Without loss of generality, suppose there are only 2 authorities (A_1, A_2) in the system. A_1 manages attributes $\{a_{1,1}, a_{1,2}\}$, and A_2 monitors attribute $\{a_{2,1}\}$. In Fig. 3, an access structure \mathcal{T} is given and marked. According to Sect. 2.4, we have $A = (1, 1), B = (2, 1), C = (2, 2), D = (3, 1), E = (3, 2)$, where A is the root node, B is the threshold node, and C, D and E are leaf nodes.

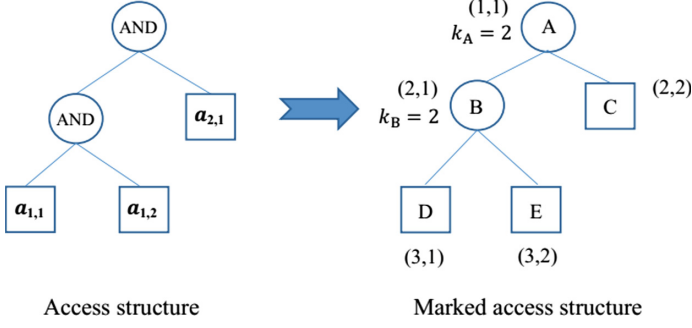


Fig. 3. Access structure and the marked access structure

For the leaf node $D = (3, 1)$, we have:

$$a_{1,1} = att(3, 1), k = 1, C_{(3,1),2} = \perp,$$

$$C_{(3,1),1} = T_{1,1}^{q(3,1)(0)} = g^{t_{1,1} \cdot q(3,1)(0)}, S_{1,1} = h^{\frac{r_1}{t_{1,1}}},$$

$$DecryptNode(CT, SK_U, (3, 1)) = e(C_{(3,1),1}, S_{1,1}) = e(g, h)^{q(3,1)(0) \cdot r_1}.$$

For the leaf node $E = (3, 2)$, we have:

$$a_{1,2} = att(3, 2), k = 1, C_{(3,2),2} = \perp,$$

$$C_{(3,2),1} = T_{1,2}^{q(3,2)(0)} = g^{t_{1,2} \cdot q(3,2)(0)}, S_{1,2} = h^{\frac{r_1}{t_{1,2}}},$$

$$DecryptNode(CT, SK_U, (3, 2)) = e(C_{(3,2),1}, S_{1,2}) = e(g, h)^{q(3,2)(0) \cdot r_1}.$$

For the leaf node $C = (2, 2)$, we have:

$$a_{2,1} = att(2, 2), k = 2, C_{(2,2),1} = \perp,$$

$$C_{(2,2),2} = T_{2,1}^{q(2,2)(0)} = g^{t_{2,1} \cdot q(2,2)(0)}, S_{2,1} = h^{\frac{r_2}{t_{2,1}}},$$

$$DecryptNode(CT, SK_U, (2, 2)) = e(C_{(2,2),2}, S_{2,1}) = e(g, h)^{q(2,2)(0) \cdot r_2}.$$

For the non-leaf node $B = (2, 1)$, We call the recursive algorithm to compute

$$\begin{aligned}
 F_{(2,1)} &= \prod_{z \in S_{(2,1)}} F_z^{\Delta_{i, S'_{(2,1)}}(0)} \\
 &= (e(g, h)^{q(2,1)(1) \cdot r_1})^{\Delta_{1, S'_{(2,1)}}} \cdot (e(g, h)^{q(2,1)(2) \cdot r_1})^{\Delta_{2, S'_{(2,1)}}} \\
 &= (e(g, h)^{r_1})^{q(2,1)(1) \Delta_{1, S'_{(2,1)}} + q(2,1)(2) \Delta_{2, S'_{(2,1)}}} \\
 &= e(g, h)^{r_1 \cdot q(2,1)(0)}
 \end{aligned}$$

where $i = index(z)$, $S'_{(x,y)} = \{index(z) : z \in S_{(x,y)}\}$, and the value of $q_{(2,1)}(0)$ is restored by Lagrange interpolation.

For the root node $A = (1, 1)$, we use the same method to calculate

$$\begin{aligned} F_{(1,1)} &= \prod_{z \in S_{(1,1)}} F_z^{\Delta_{i,S'_{(1,1)}}(0)} \\ &= (e(g, h)^{q_{(1,1)}(1) \cdot r_1})^{\Delta_{1,S'_{(1,1)}}} \cdot (e(g, h)^{q_{(1,1)}(2) \cdot r_2})^{\Delta_{2,S'_{(1,1)}}} \\ &= e(g, h)^{r_1 \cdot q_{(1,1)}(1) \Delta_{1,S'_{(1,1)}} + r_2 \cdot q_{(1,1)}(2) \Delta_{2,S'_{(1,1)}}} \end{aligned}$$

However, due to $r_1 \neq r_2$, the formula $q_{(1,1)}(1) \Delta_{1,S'_{(1,1)}} + q_{(1,1)}(2) \Delta_{2,S'_{(1,1)}}$ can not be obtained, thus the secret value $q_{(1,1)}(0)$ assigned at node A can not be restored. In this case, even if a user is legitimate, he/she cannot successfully access the data files that he/she could have accessed. It violates the intention of the data owner to encrypt data files.

In scheme [6], the same recursive operation is defined, where $Decrypt\ Node(CT, SK_U, x) = \prod_{k \in \{1, 2, \dots, N\}} e(C_{k,x}, S_{k,i}) = e(g, h)^{q_x(0) \cdot \sum r_k}$. Therefore, the scheme [6] also has the defect mentioned above.

4.3 Our Construction

Overview. In order to support correct decryption, based on the scheme [22], we made some improvements. In the KeyGen phase, we let each authority A_k sends $h^{x_{id} r_k}$ to other authorities to get the same parameter $h^{x_{id} \sum r_k}$, where x_{id} is a secret value chosen by the user, and its existence ensures that the CSS cannot get the real plaintext when performing the pre-decryption operation. The changed attribute key is $S_{k,i} = h^{\frac{x_{id} \sum r_k}{t_{k,i}}}$. However, we find that if we do not change the ciphertext component $C_m^2 = g^{s_m}$, as long as there is a corrupted authority, the user can easily get $e(g^{s_m}, h^{x_{id} \sum r_k}) = e(g, h)^{s_m \cdot x_{id} \sum r_k}$, even if his/her attributes do not meet the access policy. Therefore, the ciphertext component is changed as $C_m^2 = g^{\tau s_m}$, where $\tau \in \mathbb{Z}_p$ is a secret random number selected by the data owner. In the decryption stage, CSS provides the user with pre-decryption service, and undertakes a large amount of decryption calculations. After that, the user can obtain the corresponding plaintext through a simple calculation.

The specific scheme is constructed as follows:

The Global Setup and Authority Setup are the same as the original scheme, so we will only give a brief description here.

Global Setup. Take a security parameter λ as input, output the public parameters $PP = (e, p, g, h, \mathbb{G}, \mathbb{G}_T)$.

Authority Setup. Each authority A_k runs this algorithm and gets its public keys and secret keys:

$$PK_k = (y_k, Y_k, \{T_{k,i}\}), SK_k = (x_k, \alpha_k, \{t_{k,i}\}, \{s_{k,j}\})$$

where, $i \in \{1, 2, \dots, n_k\}, j \in \{1, 2, \dots, N\} \setminus \{k\}$.

Encryption: User first selects keys $(\kappa_1, \kappa_2, \dots, \kappa_l)$ to encrypt data files (M_1, M_2, \dots, M_l) using symmetric encryption algorithm: $C_m = Enc_{\kappa_m}(M_m)$, and then use HABE encryption algorithm to encrypt these symmetric keys as follows:

$$C_m^1 = \kappa_m \left(\prod_{k \in \{1, 2, \dots, N\}} Y_k \right)^{s_m}, C_m^2 = g^{\tau s_m}, C_{(x,y),k} = T_{k,i}^{q(x,y)(0)}$$

$$C_{(x,y),v} = \left(\prod_{k \in \{1, 2, \dots, N\}} Y_k \right)^{q(x,y)(0) + q_{child_v}(0)} \cdot H_1 \left(\left(\prod_{k \in \{1, 2, \dots, N\}} Y_k \right)^{q(x,y)(0)} \right)$$

where $\tau \in_R \mathbb{Z}_p$ is selected by encryptor, and $\tau^{-1} \bmod p$ exists.

KeyGen. The user selects a unique secret random number $x_{id} \in \mathbb{Z}_p$ and sends $h^{x_{id}}$ to the authorities. Then A_k picks $r_k \in_R \mathbb{Z}_p$ and shares $h^{x_{id} \cdot r_k}$ with other authorities. To generate the secret key for user's attribute $a_{k,i} \in \tilde{A}_U^k$, A_k calculates $S_{k,i} = h^{\frac{x_{id} \sum r_k}{t_{k,i}}}$.

The secret key D_{kj} is same as the original scheme,

$$\begin{cases} D_{kj} = g^{\alpha_k} h^{r_k} PRF_{kj}(u), k > j \\ D_{kj} = g^{\alpha_k} h^{r_k} / PRF_{kj}(u), k \leq j \end{cases}$$

User computes

$$\begin{aligned} D_U &= \prod_{(k,j) \in \{1, 2, \dots, N\} \times (\{1, 2, \dots, N\} \setminus \{k\})} D_{k,j} \\ &= g^{(N-1) \sum \alpha_k} \cdot h^{(N-1) \sum r_k} \end{aligned}$$

The user sends D_U through a secure channel to the data owner, then the data owner returns D_U^{-1} to the user.

Decryption: This decryption algorithm consists of two phases. The first stage is *CSS-Decryption*, and the second stage is *User-Decryption*.

CSS-Decryption: User sends secret keys SK_U to the CSS, which performs the following partial decryption operations.

If the node (x, y) is a leaf node and $a_{k,i} \in \tilde{A}_U^k$, then computes

$$\begin{aligned} DecryptNode(CT, SK_U, (x, y)) &= e(C_{(x,y),k}, S_{k,i}) \\ &= e(T_{k,i}^{q(x,y)(0)}, h^{\frac{x_{id} \sum r_k}{t_{k,i}}}) \\ &= e(g^{t_{k,i} q(x,y)(0)}, h^{\frac{x_{id} \sum r_k}{t_{k,i}}}) \\ &= e(g, h)^{q(x,y)(0) \cdot x_{id} \sum r_k} \end{aligned}$$

If $a_{k,i} \notin \tilde{A}_U^k$, define $DecryptNode(CT, SK_U, (x, y)) = \perp$.

The function $DecryptNode(CT, SK_U, (x, y))$ executes recursively when (x, y) is a non-leaf node. For all children nodes z for (x, y) , we call the algorithm

$DecryptNode(CT, SK, z)$ and store the output as F_z . Let $S_{(x,y)}$ be any $k_{(x,y)}$ -sized set of child nodes. If no such set exists, the function will return \perp . The recursive computation is shown as follows:

$$\begin{aligned}
F_{(x,y)} &= \prod_{z \in S_{(x,y)}} F_z^{\Delta_{i,S'_{(x,y)}}(0)} \\
&= \prod_{z \in S_{(x,y)}} (e(g, h)^{q_z(0)x_{id} \sum r_k})^{\Delta_{i,S'_{(x,y)}}(0)} \\
&= \prod_{z \in S_{(x,y)}} (e(g, h)^{q_{(x,y)}(i)x_{id} \sum r_k})^{\Delta_{i,S'_{(x,y)}}(0)} \\
&= e(g, h)^{q_{(x,y)}(0) \cdot x_{id} \sum r_k}
\end{aligned}$$

where $i = index(z)$, $S'_{(x,y)} = \{index(z) : z \in S_{(x,y)}\}$.

If user's attributes satisfy the part or whole \mathcal{T} , this algorithm continues to perform the recursive operations and gets:

$$\begin{aligned}
A_m &= DecryptNode(CT, SK_U, (x_m, y_m)) \\
&= e(g, h)^{q_{(x_m, y_m)}(0) \cdot x_{id} \sum r_k} \\
&= e(g, h)^{s_m \cdot x_{id} \sum r_k}
\end{aligned}$$

Then, CSS computes

$$B_m = e(C_m^2, D_U^{\tau-1})^{\frac{1}{N-1}} = e(g, g)^{s_m \sum \alpha_k} \cdot e(g, h)^{s_m \sum r_k}$$

Since the x_{id} is kept by the user, the CSS cannot decrypt the original message completely. Finally, the cloud server sends the result (A_m, B_m) and the ciphertext CT to the user for the next decryption calculation.

User-Decryption: User runs this algorithm to get symmetric key

$$\begin{aligned}
F_m &= \frac{B_m}{(A_m)^{\frac{1}{x_{id}}}} = \frac{e(g, g)^{s_m \sum \alpha_k} \cdot e(g, h)^{s_m \sum r_k}}{e(g, h)^{s_m \cdot \sum r_k}} = e(g, g)^{s_m \cdot \sum \alpha_k} \\
\kappa_m &= \frac{C_m^1}{F_m} = \frac{\kappa_m \cdot (\prod_{k \in \{1, 2, \dots, N\}} Y_k)^{s_m}}{e(g, g)^{s_m \cdot \sum \alpha_k}}, m \in [1, l]
\end{aligned}$$

The value of $F_{(m+1),v}$ can be obtained in the same way as the original scheme. Finally, user obtains the file M_m by using the symmetric decryption algorithm with the key κ_m .

5 Security and Performance Analysis

5.1 Security Model

The security game is played between adversary \mathcal{A} and challenger \mathcal{B} as follows:

Initialization. Adversary \mathcal{A} provides the challenger \mathcal{B} with a list of corrupted authorities $C_{\mathcal{A}}$ ($|C_{\mathcal{A}}| < N$) and an access structure \mathcal{T}^* he/she wants to challenge.

Global Setup. \mathcal{B} runs this algorithm and returns the system parameters PP to \mathcal{A} .

Authority Setup. For $A_k \in C_{\mathcal{A}}$, \mathcal{B} runs this algorithm and sends the secret-public key pair (PK_k, SK_k) to \mathcal{A} . For $A_k \notin C_{\mathcal{A}}$, \mathcal{B} only sends the public keys PK_k to \mathcal{A} .

Phase 1. \mathcal{A} provides attribute sets $\tilde{U}_1, \tilde{U}_2, \dots, \tilde{U}_q$ for q secret key queries. The only restriction is that none of these attribute sets satisfy \mathcal{T}^* . Then \mathcal{B} runs the KeyGen algorithm and outputs the corresponding secret keys.

Challenge. \mathcal{A} submits two messages M_0 and M_1 with the equal length. Then \mathcal{B} selects a random bit $b \in \{0, 1\}$ and runs the Encryption algorithm to encrypt the message M_b under the access structure \mathcal{T}^* . The corresponding ciphertext CT^* is sent to \mathcal{A} .

Phase 2. Same as phase 1.

Guess. Finally, \mathcal{A} outputs the guess b' on b . \mathcal{A} wins the game if $b' = b$.

Definition 1. A hierarchy ABE (HABE) scheme is (q, ϵ) secure against the chosen plaintext attack if all PPT adversaries making q secret key queries have the negligible advantage ϵ in the above game.

5.2 Security Analysis

Theorem 1. Our improving hierarchy CP-ABE scheme is (q, ϵ) semantically secure in the above security model, if the ϵ' -DBDH assumption holds, where

$$\epsilon' \geq \frac{\epsilon}{2} \cdot \prod_{k \in \{1, 2, \dots, N\}} \left(1 - \frac{n_k - 2}{(p-1)^2}\right).$$

Proof. Suppose there exists an adversary \mathcal{A} who can break our scheme with non-negligible advantage ϵ , then there will be a simulator \mathcal{B} who can break the DBDH assumption with advantage $\frac{\epsilon}{2} \cdot \prod_{k \in \{1, 2, \dots, N\}} \left(1 - \frac{n_k - 2}{(p-1)^2}\right)$, where n_k represents the number of attributes managed by A_k .

Comparing with scheme [22], we make some simple changes in the secret key component $S_{k,i}$ and the ciphertext component C_1^{2*} , so the simulation of these two parts needs to be changed in the security proof. Specifically, for the attribute key, if $a_{k,i} \in \mathcal{T}^*$, we replace $S_{k,i} = h^{\frac{r_k}{\omega_{k,i}}}$ with $S_{k,i} = h^{\frac{x_{id} \cdot \sum r_k}{\omega_{k,i}}}$; if $a_{k,i} \notin \mathcal{T}^*$, we replace $S_{k,i} = h^{\frac{r_k}{(a+\eta)\omega_{k,i}}}$ with $S_{k,i} = h^{\frac{x_{id} \cdot \sum r_k}{(a+\eta)\omega_{k,i}}}$. For the ciphertext component, $C_1^{2*} = g^c$ is replaced by $C_1^{2*} = g^{\tau c}$, where $\tau \in_R \mathbb{Z}_p$ is selected by the simulator \mathcal{B} .

The security proof of the proposed scheme is similar to that in scheme [22]. Due to space limitations, the complete proof is omitted here.

5.3 Performance Analysis

In Table 2, we compare the schemes [6, 22] with ours. All schemes are constructed in multi-authority system and achieve GID hiding through the anonymous key issuing protocol. However, the scheme [6] adopted the general CP-ABE approach, which is not suitable for such scenario of data encryption with hierarchical structure. In order to reduce the decryption overhead of the user, in our construction, cloud server with strong computing power performs most of the decryption calculations for the user and then sends the results to him/her. Finally, the user only needs to perform a simple calculation to obtain the final decryption result.

Table 2. Comparison of features.

Component	Qian [6]	Guo [22]	Ours
Multi-authority	Yes	Yes	Yes
Privacy-preserving	Yes	Yes	Yes
File hierarchy	No	Yes	Yes
Outsourcing	No	No	Yes
DecryptNode	$\prod_k e(C_{x,k}, S_{k,i})$	$\prod_k e(C_{(x,y),k}, S_{k,i})$	$e(C_{(x,y),k}, S_{k,i})$
Encryption time	$(A_{C_1} +1)Exp+Mul$	$(A_T A_{C_1} +l)Exp + (l+v+2) A_T Mul$	$(A_T A_{C_1} +l)Exp + (l+v+2) A_T Mul$
CSS-Decryption time	0	0	$(A_U +l)P + [(v+1) A_T + (N+1) S_1]Mul$
User-Decryption time	$(N A_U +1)P + [(N+1) S_1 +2]Mul$	$(N A_U +l)P + [2l+(v+1) A_T + (N+1) S_1]Mul$	$2lMul$

^a*Exp*: the exponential operation. *Mul*: the multiplication operation. *P*: the bilinear pairing operation. $|*|$: the number of elements in $*$.

^b A_U : the attribute set of U . A_T : the set of transport nodes. v : the number of children in $TNC(x, y)$.

For the evaluation of the computational cost of encryption and decryption, we assume that $M = (M_1, M_2, \dots, M_l)$ is the hierarchical file with l access levels and the defined access structure has l hierarchical nodes. Note that the access order is decreasing layer by layer. Let S be the least interior nodes satisfying an access structure (include the root) and A_C be the attribute set related to ciphertext CT. Thus, the attribute sets and the least interior node sets can be denoted as $\{A_{C_1}, A_{C_2}, \dots, A_{C_l}\}$, where $A_{C_1} \supset A_{C_2} \supset \dots \supset A_{C_l}$, and $\{S_1, S_2, \dots, S_l\}$, respectively.

6 Conclusions

In this paper, we first analyze Guo's scheme and find that there is a defect in the Decryption phase. Then an improving file hierarchy ABE scheme is

proposed to enable authorized users to decrypt correctly. This new scheme remains the security and privacy features of the original scheme, and reduces the user's computational overhead by handing over a large amount of decryption calculations to the cloud storage server. However, the cooperation among the multiple authorities is needed to generate the secret key for users. How to construct a decentralized HABE scheme without any cooperation is left as the future work.

References

1. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 457–473. ACM, Aarhus (2005)
2. Goyal, V., Pandey, O., Sahai, A., et al.: Attribute-based encryption for fine-grained access control of encrypted data. In: 13th ACM Conference on Computer and Communications Security, pp. 89–98. ACM, Alexandria (2006)
3. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334. IEEE, Berkeley (2007)
4. Chase, M.: Multi-authority attribute based encryption. In: 4th Conference on Theory of Cryptography, pp. 515–534. ACM, Amsterdam (2007)
5. Chase, M., Chow, S.S.M.: Improving privacy and security in multi-authority attribute-based encryption. In: 16th ACM Conference on Computer and Communications Security, pp. 121–130. ACM, Chicago (2009)
6. Qian, H., Li, J., Zhang, Y., et al.: Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation. *Int. J. Inf. Secur.* **14**(6), 487–497 (2015)
7. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: 30th Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology, pp. 568–588. ACM, Tallinn (2011)
8. Han, J., Susilo, W., Mu, Y., et al.: Privacy-preserving decentralized key-policy attribute-based encryption. *IEEE Trans. Parallel Distrib. Syst.* **23**(11), 2150–2162 (2012)
9. Ge, A., Zhang, J., Zhang, R., et al.: Security analysis of a privacy-preserving decentralized key-policy attribute-based encryption scheme. *IEEE Trans. Parallel Distrib. Syst.* **24**(11), 2319–2321 (2013)
10. Rahulamathavan, Y., Veluru, S., Han, J., et al.: User collusion avoidance scheme for privacy-preserving decentralized key-policy attribute-based encryption. *IEEE Trans. Comput.* **65**(9), 2939–2946 (2016)
11. Zhang, L., Liang, P., Mu, Y.: Improving privacy-preserving and security for decentralized key-policy attributed-based encryption. *IEEE Access* **6**, 12736–12745 (2018)
12. Qian, H., Li, J., Zhang, Y.: Privacy-preserving decentralized ciphertext-policy attribute-based encryption with fully hidden access structure. In: Qing, S., Zhou, J., Liu, D. (eds.) ICICS 2013. LNCS, vol. 8233, pp. 363–372. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-02726-5_26
13. Han, J., Susilo, W., Mu, Y., Zhou, J., Au, M.H.: PPDCP-ABE: privacy-preserving decentralized ciphertext-policy attribute-based encryption. In: Kutyłowski, M., Vaidya, J. (eds.) ESORICS 2014. LNCS, vol. 8713, pp. 73–90. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11212-1_5

14. Wang, M., Zhang, Z., Chen, C.: Security analysis of a privacy-preserving decentralized ciphertext-policy attribute-based encryption scheme. *Concurr. Comput. Practice Exp.* **28**(4), 1237–1245 (2016)
15. Yin, H., Zhang, L., Mu, Y.: A novel privacy-preserving decentralized ciphertext-policy attribute-based encryption with anonymous key generation. In: Sun, X., Pan, Z., Bertino, E. (eds.) *ICCCS 2018*. LNCS, vol. 11065, pp. 435–446. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00012-7_40
16. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) *ASIACRYPT 2002*. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36178-2_34
17. Wang, G., Liu, Q., Wu, J.: Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In: *17th ACM Conference on Computer and Communications Security*, pp. 735–737. ACM, Chicago (2010)
18. Wan, Z., Liu, J., Deng, R.H.: HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. *IEEE Trans. Inf. Forensics Secur.* **7**(2), 743–754 (2012)
19. Wang, S., Zhou, J., Liu, J.K., Yu, J., Chen, J., Xie, W.: An efficient file hierarchy attribute-based encryption scheme in cloud computing. *IEEE Trans. Inf. Forensics Secur.* **11**(6), 1265–1277 (2016)
20. Zhang, L., Wu, Q., Mu, Y., et al.: Privacy-preserving and secure sharing of PHR in the cloud. *J. Med. Syst.* **40**(12), 1–13 (2016)
21. Sandhia, G.K., Raja, S.V.K., Jansi, K.R.: Multi-authority-based file hierarchy hidden CP-ABE scheme for cloud security. *Serv. Oriented Comput. Appl.* **12**(3–4), 295–303 (2018)
22. Guo, R., Li, X., Zheng, D., et al.: An attribute-based encryption scheme with multiple authorities on hierarchical personal health record in cloud. *J. Supercomput.*, 1–20 (2018)
23. Jung, T., Li, X.Y., Wan, Z., Wan, M.: Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption. *IEEE Trans. Inf. Forensics Secur.* **10**(1), 190–199 (2015)

Public Key Cryptography



Anonymous Leakage-Resilient Ciphertext-Policy Attribute-Based Encryption Supporting Direct Revocation

Xiaoxu Gao^(✉), Leyou Zhang, and Gongcheng Hu

School of Mathematics and Statistics, Xidian University, Xi'an 710126, China
gxx_xidian@163.com

Abstract. Leakage-resilient ciphertext-policy attribute-based encryption (LR-CP-ABE) is an important tool to achieve fine-grained access control of data and resist side-channel attacks. Privacy protection and user revocation are two practical problems it faces. However, most of the existing schemes fail to achieve user revocation while protecting user's privacy at present. To address the above problems, we propose an anonymous LR-CP-ABE scheme with user revocation in this paper, which is proven to be adaptively secure in the standard model under four static assumptions over composite order group. Furthermore, we also show the proposed scheme achieves the receivers anonymity which protects the users' privacy. The performance analyses confirm the feasibility of the proposed scheme.

Keywords: Anonymous · Ciphertext-policy attribute-based encryption · Direct revocation · Leakage-resilient

1 Introduction

ABE was first proposed by Sahai and Waters [1], which is an important tool for solving security and fine-grained data sharing and access control problems. It has become a research hotspot in recent years. The ABE systems can be divided into two categories: one is CP-ABE [2] and the other is key-policy ABE (KP-ABE) [3]. The most obvious difference between them is whether the private keys are related to the attribute set. In CP-ABE, a private key is associated with an attribute list, a ciphertext is related to an access structure. The users can decrypt the ciphertexts if and only if the user's attribute set satisfies the corresponding access structure. While in KP-ABE, the situation is reversed.

Revocation is a challenge problem in the CP-ABE setting because there has opportunities to dynamically change attributes or users. Therefore, the revocation mechanism can be divided into two types, namely, attribute revocation and user revocation. So far, there are two ways to solve this problem: direct

Supported by the National Cryptography Development Fund under grant (MMJJ20180209).

revocation and indirect revocation. Indirect method means revocation mechanism by authority, which updates the private keys of a user who has not revoked an attribute periodically or dynamically. While direct method means revocation is performed by the data owner who specified the revoked user list during the encryption process. Although direct method has less flexibility in revoking users, it has an advantage in revoking costs. Revocable ABE was first proposed in [4, 5], so far, it has made great progress, such as [6–9] and so on.

Although ABE can be directly applied to the design of secure access control, for the purpose of better protecting user’s privacy and data security, anonymous ABE was proposed in [10, 11] and further improved by [12, 13]. More related works can refer to [14–19]. In anonymous ABE, the adversary cannot obtain some meaningful information about the corresponding attributes in the access policies.

However, studies have shown that these schemes can not resist various forms of attacks, such as side-channel attacks. Because the security of these schemes is based on an idealized assumption that the adversary cannot get any information of the private keys and internal state. In fact, this assumption is actually unrealistic. The adversary can learn meaningful information about the keys by using some of the physical information that the algorithm outputs. So the adversary can easily break the security of these schemes. In order to characterize the leaked information that the adversary available and protect the security of these schemes, ABE based on various leakage models are proposed in [20–27].

Zhang et al. [22] focused on the above three issues and designed a leakage-resilient secure ABE with fine-grained attribute revocation to achieve the semantic security in the continual key leakage model. Users need to pay a big price in decryption. Subsequently, Yu et al. [25] introduced a leakage-resilient CP-ABE supporting indirect revocation which can tolerate the leakage of the private keys and the master secret keys. The security of the scheme is proved by using dual system encryption.

While above schemes cannot achieve leakage-resilience, anonymity and user revocation at the same time. Therefore, it is worthwhile to study an efficient scheme that can realize the above three performances.

1.1 Our Contribution

In this paper, an CP-ABE scheme under the continuous leakage model is constructed whose leakage bound achieves $\lambda \leq (\omega - 1 - 2c) \log p_2$ during two updates, which is proved to be adaptively secure in the standard model under four static assumptions over composite order bilinear group. Moreover, this scheme can achieve the user’s direct revocation by embedding the revocation list in the ciphertexts. We also give an analysis of how the scheme achieves anonymity (Table 1).

2 Preliminaries

2.1 Linear Secret Sharing

A secret sharing scheme Λ over a set of attributes S is called linear on the two conditions that:

Table 1. Symbols

Symbol	Description
Σ	A set of attributes. In other words, $\Sigma = \{att_1, att_2, \dots, att_n\}$.
$p_{\check{i}}$	The orders of $\mathbb{G}_{p_{\check{i}}}$, where $\check{i} = 1, 2, 3, 4$
$g_{\check{i}}$	Generators of the subgroups $\mathbb{G}_{p_{\check{i}}}$ with order $p_{\check{i}}$, where $\check{i} = 1, 2, 3, 4$
\mathbb{Z}_N	The set of positive integers
pk	The public keys
msk	The master secret keys
$v_{i,j}$	The j^{th} value of att_i
sk_S	The private keys associated with attribute set $S = \{v_{1,x_1}, v_{2,x_2}, \dots, v_{n'',x_{n''}}\}$
m	Messages
CT	The ciphertexts
$x \in_R X$	Denote that x is randomly chosen from a set X
\mathbf{A}	A matrix
\mathbf{v}	A vector
$[n]$	A set of values from 1 to n
n_i	The possible values of the attribute att_i

- (1) The shares for each attributes form a vector from \mathbb{Z}_p .
- (2) There exists a $l \times n$ matrix \mathbf{A} called sharing-generating matrix for Λ . The function ρ maps the x^{th} row of \mathbf{A} to an attribute value labeling $\rho(x)$ for all $x \in [l]$. Then we selects a vector $\mathbf{v} = (s, v_2, \dots, v_n) \in_R \mathbb{Z}_p^n$, where s is the secret to be shared, and $\mathbf{A} \cdot \mathbf{v}$ is the vector of l shares of the secret s according to Λ . The shares $(\mathbf{A}\mathbf{v})_x$ belongs to the attribute value $\rho(x)$.

Linear Reconstruction. Let $C \in \Lambda$ be any authorized set, and let $I \subset \{1, 2, \dots, l\}$ be defined as $I = \{x' | \rho(x') \in C\}$. Then, there exists constants $\{\mu_{x'} \in \mathbb{Z}_p\}_{x' \in I}$ such that, if $\{\lambda_{x'}\}$ are valid shares of any s in Λ , then $\sum_{x' \in I} \mu_{x'} \lambda_{x'} = s$. This collection $\{\mu_{x'}\}_{x' \in I}$ can be found in polynomial time.

2.2 Complexity Assumptions

Assumption 1. Given a instance $(\Theta = (N = p_1 p_2 p_3 p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_1, g_3, g_4, T)$, where $g_{\check{i}} \in_R \mathbb{G}_{p_{\check{i}}}$ for $\check{i} = 1, 3, 4$, the advantage of \mathcal{A} distinguish $T \in_R \mathbb{G}_{p_1 p_4}$ from $T \in_R \mathbb{G}_{p_1 p_2 p_4}$ is negligible.

Assumption 2. Given instance $(\Theta = (N = p_1 p_2 p_3 p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_1, g_3, g_4, U_1 U_2, W_2 W_3, T)$, where $g_1, U_1 \in_R \mathbb{G}_{p_1}$, $U_2, W_2 \in_R \mathbb{G}_{p_2}$, $g_3, W_3 \in_R \mathbb{G}_{p_3}$ and $g_4 \in_R \mathbb{G}_{p_4}$, the advantage of \mathcal{A} distinguish $T \in_R \mathbb{G}_{p_1 p_3}$ from $T \in_R \mathbb{G}_{p_1 p_2 p_3}$ is negligible.

Assumption 3. Given a instance $(\Theta = (N = p_1 p_2 p_3 p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_1, g_2, g_3, g_4, g_1^\alpha U_2, g_1^s W_2, g_2^r, U_2^r, T)$, where $s, \alpha, r \in_R \mathbb{Z}_N$, $g_1 \in_R \mathbb{G}_{p_1}$, $g_2, U_2, W_2 \in_R \mathbb{G}_{p_2}$ and $g_3 \in_R \mathbb{G}_{p_3}$, the advantage of \mathcal{A} distinguish $T = \hat{e}(g, g)^{\alpha s}$ from $T \in_R \mathbb{G}_T$ is negligible.

Assumption 4. Given a instance $(\Theta = (N = p_1 p_2 p_3 p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_1, g_2, g_3, g_4, U_1 U_4, U_1^{\hat{r}} U_2, g_1^{\hat{s}} W_2, g_1^s W_{24}, U_1 g_3^{\hat{s}}, T)$, where $s, \hat{r}, \hat{s} \in_R \mathbb{Z}_N$, $g_1, U_1 \in_R \mathbb{G}_{p_1}$, $g_2, U_2, W_2 \in_R \mathbb{G}_{p_2}$, $g_3 \in_R \mathbb{G}_{p_3}$, $g_4, U_4 \in_R \mathbb{G}_{p_4}$ and $W_{24}, D_{24} \in_R \mathbb{G}_{p_2 p_4}$, the advantage of \mathcal{A} distinguish $T \in_R U_1^s D_{24}$ from $T \in_R \mathbb{G}_{p_1 p_2 p_4}$ is negligible.

2.3 Random Subspaces for Leakage Resilience over Arbitrary Functions

Theorem 1. For any function $f : \mathbb{Z}_p^{m' \times d'} \rightarrow \phi$, there exists

$$\text{Dist}((\mathbf{X}_1, f(\mathbf{X}_1 \mathbf{T})), (\mathbf{X}_1, f(\mathbf{X}_2))) \leq \epsilon,$$

where $m', l', d' \in_R \mathbb{N}$, $2d' \leq l' \leq m'$, $\mathbf{X}_1 \in_R \mathbb{Z}_p^{m' \times l'}$, $\mathbf{X}_2 \in_R \mathbb{Z}_p^{m' \times d'}$, $\mathbf{T} \in_R \text{Rank}_{d'}(\mathbb{Z}_p^{l' \times d'})$, $|\phi| \leq 4(1 - \frac{1}{p}) \cdot p_2^{l' - 2d' + 1} \cdot \epsilon^2$.

Claim. For any function $f : \mathbb{Z}_p^{m'} \rightarrow \{0, 1\}^{l'}$, there exists

$$\text{Dist}((\Delta, f(\boldsymbol{\mu})), (\Delta, f(\boldsymbol{\mu}'))) \leq \epsilon,$$

where $\Delta, \boldsymbol{\mu} \in_R \mathbb{Z}_p^{m'}$, $\boldsymbol{\mu}' \cdot \Delta = 0 \pmod{p}$, $l' \leq 4p^{m' - 3}(p - 1) \cdot \epsilon^2$.

3 LR-CP-ABE Supporting Direct Revocation

3.1 Model of LR-CP-ABE with Direct Revocation

Three entities are included in our construction: attribute authority (AA), data owners (DO) and users.

Setup $(\kappa, \Sigma, \lambda)$: AA takes the security parameter κ , universe attribute set Σ and leakage bound λ as input, outputs the public keys pk and master secret keys msk .

KeyGen (pk, msk, S, id) : AA inputs the public keys pk , master secret keys msk , attribute list S for the user with id , outputs the private keys sk_S .

UpdateUsk (pk, sk_S) : AA takes the public keys pk and the secret keys sk_S as input, outputs the new private keys sk'_S .

Encrypt $(pk, m, \Lambda, \mathcal{R})$: DO takes the public keys pk , a message m , access structure Λ and revocation list \mathcal{R} as input, then outputs the ciphertexts CT .

Decrypt (CT, sk_S) : The users inputs the ciphertexts CT and the private keys sk_S , and outputs the message m .

3.2 Security Properties of the ANON-LR-CP-ABE with Direct Revocation

This game is played by the interaction between an adversary \mathcal{A} and a challenger \mathcal{C} , the concrete process is described as follows:

- **Setup:** \mathcal{C} inputs the security parameter κ and the leakage upper bound λ , generates the public keys pk and the master secret keys msk . Then \mathcal{C} sends pk to \mathcal{A} while keeps msk . At the same time, \mathcal{C} creates an initial empty lists: $\mathcal{L} = (hd, S, sk_S, L_{sk})$, where L_{sk} means the total leakage bits.
- **Phase 1:** \mathcal{A} adaptively performs the following queries:
 - *KeyGen queries:* \mathcal{A} sends an identity id and an attribute list S to \mathcal{C} , then \mathcal{C} runs the algorithm **KeyGen** to generate the private keys sk_S . Finally, \mathcal{C} updates $hd = hd + 1$ and adds the item $(hd, S, sk_S, 0)$ to the list \mathcal{L} .
 - *Leakage queries:* \mathcal{A} gives a polynomial-time computable arbitrary function $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ to \mathcal{C} . Assume that the set is (hd, S, sk_S, L_{sk_S}) , then \mathcal{C} checks whether $|f(sk_S)| + L_{sk_S} \leq \lambda$. If this is true, it returns $f(sk_S)$ to \mathcal{A} . Otherwise, outputs the symbol \perp .
 - *UpdateUsk queries:* \mathcal{A} queries the new updated secret keys for hd . If there is no (hd, S, sk_S, L_{sk_S}) found in set \mathcal{L} . Then \mathcal{C} runs the algorithm **KeyGen** to get the private keys sk_S and sets $L_{sk_S} = 0$. Otherwise, \mathcal{C} returns re-randomized private keys sk'_S with **UpdatedUsk** and updates the corresponding $L_{sk_S} = 0$.
- **Challenge:** \mathcal{A} outputs two messages of the same length m_0, m_1 , revocation list \mathcal{R} and two challenge access structures $A_0(\mathbf{A}_0, \rho_0)$, $A_1(\mathbf{A}_1, \rho_1)$ to \mathcal{C} , then \mathcal{C} selects $b \in \{0, 1\}$ randomly and encrypts the message m_b under the access structure $A_b(\mathbf{A}_b, \rho_b)$. Finally, it outputs the ciphertexts CT^* to \mathcal{A} .
- **Phase 2:** The phase is similar to **Phase 1** except that \mathcal{A} cannot execute the *Leakage queries* and the *KeyGen queries* that the corresponding attribute set satisfies the challenge access structure.
- **Guess:** \mathcal{A} outputs the guess b' of b and wins the game if $b' = b$.

If the advantage of \mathcal{A} in the above game is negligible, then it is said that the anonymous CP-ABE scheme which supporting direct revocation is indistinguishable under the chosen plaintext attack (ANON-IND-CPA-REVO) and it is λ leakage-resilient, where the advantage of \mathcal{A} is defined as

$$Adv_{\mathcal{A}}^{ANON-IND-CPA-REVO} = |\Pr[b' = b] - \frac{1}{2}|$$

4 Construction

4.1 Concrete Construction

Setup(κ, Σ, λ): AA takes a security parameter κ and the attribute universe description Σ and a leakage bound λ as input. Then it runs the bilinear group generator to produce $\Theta = (N = p_1 p_2 p_3 p_4, \mathbb{G}, \mathbb{G}_T, \hat{e})$, defines $negl = p_2^{-c}$ as the

allowable maximum probability in succeeding in leakage guess and computes $\omega = \lceil 1 + 2c + \frac{\lambda}{\log p_2} \rceil$, where c is a positive constant. Then the algorithm generates the public keys as follows. First, it selects $g_1, h \in \mathbb{G}_{p_1}$, $g_3 \in \mathbb{G}_{p_3}$ and $a, \alpha \in \mathbb{Z}_N$ at random. Second, it selects $\rho \in_R \mathbb{Z}_N^\omega$ and selects $t_{i,j} \in_R \mathbb{Z}_N$, $g_4, w_0, w_{i,j} \in_R \mathbb{G}_{p_4}$ for each $i \in [n], j \in [n_i]$, the public keys are

$$pk = \left(N, a_0, h, u, g_3, g_1^\rho, y, T_{i,j}; \forall i \in [n], j \in [n_i] \right)$$

where $a_0 = g_1 w_0, u = g_1^\alpha g_4, y = e(g_1, g_1)^\alpha, T_{i,j} = g_1^{t_{i,j}} w_{i,j}$.

The master secret keys are

$$msk = (a, \alpha, t_{i,j}, g_1).$$

KeyGen(pk, msk, S, id): On input the public keys pk , the master keys msk , an attribute set S and users identity id , AA outputs the secret keys $sk_S = \left(S, sk_{S,1}, sk_{S,2} \right) = \left(S, \{\mathbf{k}_0, k_1\}, \{k_{2,i}, k_{3,i}, k_{4,i}\}_{i \in S} \right)$ and sends them to users. AA selects $r_{id}, y_1 \in_R \mathbb{Z}_N$, $\mathbf{y}_0, \sigma \in_R \mathbb{Z}_N^\omega$ and picks $r_{i,j}, y_{i,j,2}, y_{i,j,3}, y_{i,j,4} \in_R \mathbb{Z}_N$ for $v_{i,j} \in S$, calculates and outputs the secret keys as follows.

$$\begin{aligned} sk_S &= \left(S, sk_{S,1}, sk_{S,2} \right) \\ &= \left(S, \{\mathbf{k}_0, k_1\}, \{k_{i,j,2}, k_{i,j,3}, k_{i,j,4}\}_{v_{i,j} \in S} \right) \\ &= \left(S, \{g_1^\sigma * g_3^{\mathbf{y}_0}, g_1^{\alpha + ar_{id} + \langle \sigma, \rho \rangle} y_1\}, \{g_1^{\alpha r_{id} + t_{i,j} r_{i,j} + ar_{i,j}} g_3^{y_{i,j,2}}, g_1^{r_{i,j}} g_3^{y_{i,j,3}}, \right. \\ &\quad \left. (g_1^{aid} h)^{r_{i,j}} g_3^{y_{i,j,4}}\}_{v_{i,j} \in S} \right) \end{aligned} \tag{1}$$

UpdateSk(sk_S, S): AA selects $\Delta r_{id}, \Delta y_1 \in_R \mathbb{Z}_N$, $\Delta \sigma, \Delta \mathbf{y}_0 \in_R \mathbb{Z}_N^\omega$ and $\Delta r_{i,j}, \Delta y_{i,j,2}, \Delta y_{i,j,3}, \Delta y_{i,j,4} \in_R \mathbb{Z}_N$ for $v_{i,j} \in S$, outputs the re-randomized keys sk'_S :

$$\begin{aligned} sk'_S &= \left(S, sk'_{S,1}, sk'_{S,2} \right) \\ &= \left(S, \{\mathbf{k}'_0, k'_1\}, \{k'_{i,j,2}, k'_{i,j,3}, k'_{i,j,4}\}_{v_{i,j} \in S} \right) \\ &= \left(S, \{\mathbf{k}_0 * g_1^{\Delta \sigma} * g_3^{\Delta \mathbf{y}_0}, k_1 g_1^{a \Delta r_{id} + \langle \rho, \Delta \sigma \rangle} g_3^{\Delta y_1}\}, \{k_{i,j,2} g_1^{\alpha \Delta r_{id} + t_{i,j} \Delta r_{i,j} + a \Delta r_{i,j}} \right. \\ &\quad \left. g_3^{\Delta y_{i,j,2}}, k_{i,j,3} g_1^{\Delta r_{i,j}} g_3^{\Delta y_{i,j,3}}, k_{i,j,4} (g_1^{aid} h)^{\Delta r_{i,j}} g_3^{\Delta y_{i,j,4}}\}_{v_{i,j} \in S} \right) \end{aligned} \tag{2}$$

Encrypt(pk, m, Λ): \mathbf{A} in $\Lambda(\mathbf{A}, \rho)$ is a secret sharing matrix of $l \times n$, where ρ maps rows of \mathbf{A} into attribute values. $\mathcal{R} = \{R_{\rho(x)}\}_{x \in [l]}$ be an attribute

revocation list. DO selects $\mathbf{v} = (s, v_2, \dots, v_n) \in \mathbb{Z}_N^n$ at random. The revocation list of attribute $\rho(x)$ is $R_{\rho(x)} = \{id_1, id_2, \dots, id_{l_x}\}$, where l_x is a variable number of revocation users. Then the algorithm selects $s_{x,i'} \in_R \mathbb{Z}_N$ for each $id_{i'} \in R_{\rho(x)}$ with the restriction that $\sum_{i'=1}^{l_x} s_{x,i'} = \lambda_x$ where $\lambda_x = \mathbf{A}_x \cdot \mathbf{v}$, $g_4, w_1, w_{\lambda_x,1}, w_{\lambda_x,2}, w_{x,i',1}, w_{x,i',2} \in_R \mathbb{G}_{p_4}$, \mathbf{A}_x is the x^{th} row of \mathbf{A} . Finally, the algorithm outputs the ciphertexts CT as follows:

$$\begin{aligned}
 CT &= \left(\mathbf{A}, \{I_x\}_{x \in [l]}, \mathcal{R}, c_0, \mathbf{c}_1, c_2, \{c_{x,0}, c_{x,1}, \{c_{x,i'}^1, c_{x,i'}^2\}_{i' \in \{1,2,\dots,l_x\}}\}_{x \in [l]} \right) \\
 &= \left(\mathbf{A}, \{I_x\}_{x \in [l]}, \mathcal{R}, m y^s, a_0^{-s\rho} * g_4^\mu, a_0^s \cdot w_1, \{a_0^{\lambda_x} \cdot w_{\lambda_x,1}, T_{\rho(x)}^{\lambda_x} \cdot w_{\lambda_x,2}, \right. \\
 &\quad \left. \{a_0^{s_{x,i'}} \cdot w_{x,i',1}, (u^{id_{i'}} h)^{s_{x,i'}} \cdot w_{x,i',2}\}_{i' \in \{1,2,\dots,l_x\}}\}_{x \in [l]} \right) \tag{3}
 \end{aligned}$$

where $\boldsymbol{\mu} \in_R \mathbb{Z}_N^\omega$, $\{I_x\}_{x \in [l]} \subset \{1, 2, \dots, n\}$ is the index set of corresponding attribute name.

Decrypt(CT, sk_S): This algorithm takes the public keys pk , users identity id , the ciphertexts CT and the secret keys sk_S as input. If $id \in R_{\rho(x)}$, then the algorithm aborts. Otherwise, suppose $\mathcal{H} = \{x | \rho(x) \in S, id \notin R_{\rho(x)}\}$. If $S' = \{\rho(x) | x \in \mathcal{H}\}$ satisfies the access structure, then users computes $d_{x,1}, d_{x,2}$ for every $x \in \mathcal{H}$ at first.

$$\begin{aligned}
 d_{x,1} &= \prod_{i'=1}^{l_x} \left(\frac{\hat{e}(k_{\rho(x),3}, c_{x,i'}^2)}{\hat{e}(k_{\rho(x),4}, c_{x,i'}^1)} \right)^{\frac{1}{id-id_{i'}}} \\
 &= \prod_{i'=1}^{l_x} \left(\frac{\hat{e}(g_1^{r_{\rho(x)}} g_3^{y_{\rho(x),3}}, (u^{id_{i'}} h)^{s_{x,i'}})}{\hat{e}((g^{aid} h)^{r_{\rho(x)}} g_3^{y_{\rho(x),4}}, g_1^{s_{x,i'}})} \right)^{\frac{1}{id-id_{i'}}} \tag{4}
 \end{aligned}$$

$$\begin{aligned}
 &= \prod_{i'=1}^{l_x} \hat{e}(g_1, g_1)^{-a r_{\rho(x)} s_{x,i'}} \\
 &= \hat{e}(g_1, g_1)^{-a \lambda_x r_{\rho(x)}} \\
 d_{x,2} &= \frac{\hat{e}(k_{\rho(x),2}, c_{x,0})}{\hat{e}(k_{\rho(x),3}, c_{x,1})} \\
 &= \frac{\hat{e}(g_1^{\alpha r_{id} + t_{\rho(x)} r_{\rho(x)} + a r_{\rho(x)}} g_3^{y_{\rho(x),2}}, g^{\lambda_x})}{\hat{e}(g_1^{r_{\rho(x)}} g_3^{y_{\rho(x),3}}, T_{\rho(x)}^{\lambda_x})} \tag{5} \\
 &= \hat{e}(g_1, g_1)^{\alpha r_{id} \lambda_x + a \lambda_x r_{\rho(x)}}
 \end{aligned}$$

Obvious, there are

$$\begin{aligned}
 d_x &= d_{x,1} d_{x,2} = \hat{e}(g_1, g_1)^{\alpha r_{id} \lambda_x} \\
 CT' &= \prod_{x \in \mathcal{H}} d_x^{\mu_x} = \hat{e}(g_1, g_1)^{\alpha r_{id} s} \tag{6}
 \end{aligned}$$

where $\sum_{x \in \mathcal{H}} \mu_x \mathbf{A}_x = (1, 0, 0, \dots, 0)$. Finally, it computes the $m = \frac{c_0}{\hat{e}_\omega(\mathbf{c}_1, \mathbf{k}_0) \hat{e}(c_2, \mathbf{k}_1)} CT'$.

4.2 Security Proof

The security proof is based on dual system encryption, so we define the semi-functional keys and semi-functional ciphertexts as follows:

Semi-functional Keys: There are two types of semi-functional keys in our proof. Firstly, we run the **KeyGen** to get normal private keys as: $sk_S = \left(S, sk_{S,1}, sk_{S,2} \right) = \left(S, \{k'_0, k'_1\}, \{k'_{i,j,2}, k'_{i,j,3}, k_{i,j,4}\}_{v_{i,j} \in S} \right)$. Then it selects $d_0 \in_R \mathbb{Z}_N^\omega, d_1 \in_R \mathbb{Z}_N$ and $d_{i,j,2}, d_{i,j,3}, d_{i,j,4} \in_R \mathbb{Z}_N$ for $v_{i,j} \in S$ and compute two types of semi-functional private keys components as follows.

Type 1.

$$\begin{aligned} k_0 &= k'_0 * g_2^{d_0}, & k_1 &= k'_1 g_2^{d_1}, & k_{i,j,2} &= k'_{i,j,2} g_2^{d_{i,j,2}}, \\ k_{i,j,3} &= k'_{i,j,3} g_2^{d_{i,j,3}}, & k_{i,j,4} &= k'_{i,j,4} g_2^{d_{i,j,4}}. \end{aligned}$$

Type 2.

$$\begin{aligned} k_0 &= k'_0 * g_2^{d_0}, & k_1 &= k'_1 g_2^{d_1}, & k_{i,j,2} &= k'_{i,j,2} g_2^{d_{i,j,2}}, \\ k_{i,j,3} &= k'_{i,j,3}, & k_{i,j,4} &= k'_{i,j,4}. \end{aligned}$$

Semi-functional Ciphertexts: For an access structure $\Lambda(\mathbf{A}, \rho)$ and a revocation list \mathcal{R} , we first run the encryption algorithm **Encrypt** to obtain normal ciphertexts $CT = \left(\mathbf{A}, \{I_x\}_{x \in [l]}, \mathcal{R}, c_0, c'_1, c'_2, \{c'_{x,0}, c'_{x,1}, \{c_{x,i'}^1, c_{x,i'}^2\}_{i' \in \{1,2,\dots,l_x\}}\}_{x \in [l]} \right)$ and choose some random elements $e_1 \in \mathbb{Z}_N^\omega$ and $e_2, e_{x,0}, e_{x,1}, e_{x,i',1}, e_{x,i',2} \in \mathbb{Z}_N$. The semi-functional ciphertexts are computed as follows:

$$\begin{aligned} c_1 &= c'_1 * g_2^{e_1}, & c_2 &= c'_2 g_2^{e_2}, & c_{x,0} &= c'_{x,0} g_2^{e_{x,0}}, \\ c_{x,1} &= c'_{x,1} g_2^{e_{x,1}}, & c_{x,i'}^1 &= c_{x,i'}^1 g_2^{e_{x,i',1}}, & c_{x,i'}^2 &= c_{x,i'}^2 g_2^{e_{x,i',2}}. \end{aligned}$$

The security of the program is proved by a series of indistinguishable games. The specific game definitions are described below:

Game_{real}: This is a real game that the private keys and ciphertexts are in normal form.

Game₀: The game is similar to the *Game_{real}* except that the ciphertexts are semi-functional.

Game_{k-1,2}: The first $k - 1$ private keys are semi-functional of **Type 2**, the rest of private keys are normal.

Game_{k,1}: The game is similar to the *Game_{k-1,2}* except the k^{th} private key is semi-functional of **Type 1**.

Game_{k,2}: The game is similar to the *Game_{k,1}* except the k^{th} private key is semi-functional of **Type 2**.

$Game_{q,2}$: All of private keys are semi-functional of **Type 2** and the ciphertexts are semi-functional, where q is the number of queries.

$Game_{final,0}$: The ciphertext component c_0 is the encryption of a random message.

$Game_{final,1}$: The component $c_{x,i',2}$ is a random element in subgroup $\mathbb{G}_{p_1 p_2 p_4}$.

Lemma 1. *Suppose that there is an adversary \mathcal{A} can distinguish the $Game_{real}$ and $Game_0$ with a non-negligible advantage ϵ , then there is a simulator \mathcal{B} breaks the Assumption 1 with same advantage.*

Proof. \mathcal{B} receives the challenge instance $(\Theta = (N = p_1 p_2 p_3 p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_1, g_3, g_4, T)$ from the challenge \mathcal{C} and simulates the $Game_{real}$ or $Game_0$.

Setup: After receiving the challenge instance $(\Theta = (N = p_1 p_2 p_3 p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_1, g_3, g_4, T)$, \mathcal{B} generates the public keys as follows: $pk = \left(N, a_0 = g_1 g_4^{\alpha'}, h = g_1^t, u = g_1^{\alpha} g_4, g_3, g_1^{\rho}, y = e(g_1, g_1)^{\alpha}, T_{i,j} = g_1^{t_{i,j}} g_4^{a_{i,j}}, \forall i \in [n], j \in [n_i] \right)$, where $t, a, a', \alpha, t_{i,j}, a_{i,j} \in_R \mathbb{Z}_N, \rho \in_R \mathbb{Z}_N^{\omega}$.

Phase 1: Because \mathcal{B} knows the master keys, so it can answer all the *KeyGen queries* and *Leakage queries*.

Challenge: \mathcal{A} sends two challenge access structure $\Lambda_0^*(\mathbf{A}_0^*, \rho_0^*), \Lambda_1^*(\mathbf{A}_1^*, \rho_1^*)$, two messages m_0, m_1 of equal length and a revocation list $\mathcal{R} = \{R_{\rho(x)}\}_{x \in [l]}$ to \mathcal{B} , then \mathcal{B} selects $b \in \{0, 1\}$ at random and computes the ciphertexts as follows: $CT = \left(\mathbf{A}_b^*, \{I_{b,x}\}_{x \in [l]}, \mathcal{R}, c_0 = m_b \hat{e}(T, g)^{\alpha}, c'_1 = T^{-\rho} g_4^{\alpha}, c'_2 = T g_4^{w'_1}, \{c'_{x,0} = T^{\lambda_{b,x}} g_4^{w'_{\lambda_{b,x},1}}, c'_{x,1} = T^{t_{\rho(x)} \lambda_{b,x}} g_4^{w'_{\lambda_{b,x},2}}, \{c'_{x,i'} = T^{s_{x,i'}} g_4^{w'_{x,i',1}}, c'_{x,i'} = T^{(a i' + t) s_{x,i'}} g_4^{w'_{x,i',2}}\}_{i' \in \{1,2,\dots,l_x\}}\}_{x \in [l]} \right)$, where $\lambda_{b,x} = \mathbf{A}_{b,x}^* \cdot \mathbf{v}', \mathbf{u}, \mathbf{v}' = (1, v'_2, v'_3, \dots, v'_n) \in_R \mathbb{Z}_N^{\omega}, \sum_{i'=1}^{l_x} s_{x,i'} = \lambda_{b,x}, w'_1, w'_{\lambda_{b,x},1}, w'_{\lambda_{b,x},2}, w'_{x,i',1}, w'_{x,i',2}, s_{x,i'} \in_R \mathbb{Z}_N, \{I_x\}_{x \in [l]}$ is the index set of corresponding attribute name.

Phase 2: Same as **Phase 1** except that \mathcal{A} cannot execute the *Leakage queries* and *KeyGen queries* that the corresponding attribute set satisfies the challenge access structure.

Guess: \mathcal{A} outputs the guess of b' of b . If $b' = b$, \mathcal{A} wins the game.

If $T \in_R \mathbb{G}_{p_1 p_4}$, then \mathcal{B} simulates the $Game_{real}$. Otherwise, \mathcal{B} simulates the $Game_0$. Therefore, if \mathcal{A} can distinguish these two games with a non-negligible advantage, then \mathcal{B} can break the Assumption 1 with same advantage.

Lemma 2. *Suppose that there is an adversary \mathcal{A} can distinguish the $Game_{k-1,2}$ and $Game_{k,1}$ with a non-negligible advantage ϵ , then there is a simulator \mathcal{B} breaks the Assumption 2 with same advantage.*

Proof. \mathcal{B} receives the challenge instance $(\Theta = (N = p_1 p_2 p_3 p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_1, g_3, g_4, U_1 U_2, W_2 W_3, T)$ from the challenge \mathcal{C} and simulates the $Game_{k-1,2}$ or $Game_{k,1}$.

Setup: The algorithm of **Setup** is same as that in Lemma 1.

KeyGen queries in Phase 1: To generate the first $k-1$ semi-functional keys, \mathcal{B} chooses $r_{id}, y_1 \in \mathbb{Z}_N$ at random, $\mathbf{y}_0, \boldsymbol{\sigma} \in_R \mathbb{Z}_N^\omega$ and $r_{i,j}, y_{i,j,2}, y_{i,j,3}, y_{i,j,4} \in_R \mathbb{Z}_N$ for $v_{i,j} \in S$, calculates and outputs the secret keys of **Type 1** as follows.

$$\begin{aligned} sk_S &= \left(S, sk_{S,1}, sk_{S,2} \right) \\ &= \left(S, \{\mathbf{k}_0, \mathbf{k}_1\}, \{k_{i,j,2}, k_{i,j,3}, k_{i,j,4}\}_{v_{i,j} \in S} \right) \\ &= \left(S, \{g_1^{\boldsymbol{\sigma}} * (W_2 W_3)^{\mathbf{y}_0}, g_1^{\alpha ar_{id} + \langle \boldsymbol{\sigma}, \boldsymbol{\rho} \rangle} (W_2 W_3)^{y_1}\}, \{g_1^{\alpha r_{id} + t_{i,j} r_{i,j} + ar_{i,j}} \right. \\ &\quad \left. (W_2 W_3)^{y_{i,j,2}}, g_1^{r_{i,j}} g_3^{y_{i,j,3}}, (g_1^{aid} h)^{r_{i,j}} g_3^{y_{i,j,4}}\}_{v_{i,j} \in S} \right) \end{aligned}$$

To generate the k^{th} private key, \mathcal{B} picks $r_{id}, y_1 \in \mathbb{Z}_N$ randomly, $\mathbf{y}_0, \boldsymbol{\sigma}' \in_R \mathbb{Z}_N^\omega$ and $r_{i,j}, y_{i,j,2}, y_{i,j,3}, y_{i,j,4} \in_R \mathbb{Z}_N$ for $v_{i,j} \in S$, outputs the following secret keys .

$$\begin{aligned} sk_S &= \left(S, sk_{S,1}, sk_{S,2} \right) \\ &= \left(S, \{\mathbf{k}_0, \mathbf{k}_1\}, \{k_{i,j,2}, k_{i,j,3}, k_{i,j,4}\}_{v_{i,j} \in S} \right) \\ &= \left(S, \{T^{\boldsymbol{\sigma}'} * g_3^{\mathbf{y}_0}, g_1^{\alpha T a + \langle \boldsymbol{\sigma}', \boldsymbol{\rho} \rangle} g_3^{y_1}\}, \{T^\alpha g_1^{t_{i,j} r_{i,j} + ar_{i,j}} g_3^{y_{i,j,2}}, \right. \\ &\quad \left. g_1^{r_{i,j}} g_3^{y_{i,j,3}}, (g_1^{aid} h)^{r_{i,j}} g_3^{y_{i,j,4}}\}_{v_{i,j} \in S} \right) \end{aligned}$$

The rest of private keys are normal keys.

Challenge: \mathcal{A} sends two challenge access structure $\Lambda_0^*(\mathbf{A}_0^*, \rho_0^*), \Lambda_1^*(\mathbf{A}_1^*, \rho_1^*)$, two message m_0, m_1 of equal length and a revocation list $\mathcal{R} = \{R_{\rho(x)}\}_{x \in [l]}$ to \mathcal{B} , then \mathcal{B} selects $b \in \{0, 1\}$ at random and calculates the ciphertexts as follows:

$$CT = \left(\mathbf{A}_b^*, \{I_{b,x}\}_{x \in [l]}, \mathcal{R}, c_0 = m_b \hat{e}(U_1 U_2, g)^\alpha, c'_1 = (U_1 U_2)^{-\rho} g_4^u, c'_2 = (U_1 U_2) g_4^{w'_1}, \{c'_{x,0} = (U_1 U_2)^{\lambda_{b,x}} g_4^{w'_{\lambda_{b,x},1}}, c'_{x,1} = (U_1 U_2)^{t_{\rho(x)} \lambda_{b,x}} g_4^{w'_{\lambda_{b,x},2}}, \{c'_{x,i'} = (U_1 U_2)^{s_{x,i'}} g_4^{w'_{s_{x,i'},1}}, c'_{x,i'} = ((U_1 U_2)^{(aid_{i'} + t) s_{x,i'}} g_4^{w'_{s_{x,i'},2}}\}_{i' \in \{1, 2, \dots, l_x\}}\}_{x \in [l]} \right),$$

where $\lambda_{b,x} = \mathbf{A}_x^* \cdot \mathbf{v}', \mathbf{u}, \mathbf{v}' = (1, v'_2, v'_3, \dots, v'_n) \in_R \mathbb{Z}_N^\omega, \sum_{i'=1}^{l_x} s_{x,i'} = \lambda_{b,x}, w'_1, w'_{\lambda_{b,x},1}, w'_{\lambda_{b,x},2}, w'_{s_{x,i'},1}, w'_{s_{x,i'},2}, s_{x,i'} \in_R \mathbb{Z}_N, \{I_x\}_{x \in [l]}$ is the index set of corresponding attribute name.

Phase 2: Same as **Phase 2** in Lemma 1.

Guess: \mathcal{A} outputs the guess of b' of b . If $b' = b$, \mathcal{A} wins the game.

It can be learn from the analysis above that \mathcal{B} simulates the $Game_{k-1,2}$ if $T \in_R \mathbb{G}_{p_1 p_3}$. Vice versa. So if \mathcal{A} distinguish these two games with a non-negligible advantage ϵ , then there is a simulator \mathcal{B} break the Assumption 2 with same advantage.

Lemma 3. *Suppose that there is an adversary \mathcal{A} can distinguish the $Game_{k,1}$ and $Game_{k,2}$ with a non-negligible advantage ϵ , then there is a simulator \mathcal{B} breaks the Assumption 2 with same advantage.*

Proof. \mathcal{B} receives the challenge instance $(\Theta = (N = p_1 p_2 p_3 p_4, \mathbb{G}, \mathbb{G}_T, \hat{e}), g_1, g_3, g_4, U_1 U_2, W_2 W_3, T)$ from the challenge \mathcal{C} and simulates the $Game_{k,1}$ or $Game_{k,2}$.

The proof of Lemma 3 is similar to that of Lemma 2 except the construction of k^{th} private key.

$$\begin{aligned} sk_S &= \left(S, sk_{S,1}, sk_{S,2} \right) \\ &= \left(S, \{k_0, k_1\}, \{k_{i,j,2}, k_{i,j,3}, k_{i,j,4}\}_{v_{i,j} \in S} \right) \\ &= \left(S, \{g_1^{\sigma'} * (W_2 W_3)^{y_0}, g_1^{\alpha + a + (\sigma' \cdot \rho)} (W_2 W_3)^{y_1}\}, \{g_1^{\alpha r_{id}} T^{t_{i,j} + a} (W_2 W_3)^{y_{i,j,2}}, \right. \\ &\quad \left. T g_3^{y_{i,j,3}}, T^{aid+t} g_3^{y_{i,j,4}}\}_{v_{i,j} \in S} \right) \end{aligned}$$

If $T \in_R \mathbb{G}_{p_1 p_2 p_3}$, then \mathcal{B} simulates the $Game_{k,1}$. Otherwise, \mathcal{B} simulates the $Game_{k,2}$. So if \mathcal{A} can distinguish these two schemes with a non-negligible advantage, then there is a simulator \mathcal{B} breaks the Assumption 2 with same advantage.

Lemma 4. *Suppose that there is an adversary \mathcal{A} can distinguish the $Game_{q,2}$ and $Game_{final,0}$ with a non-negligible advantage ϵ , then there is a simulator \mathcal{B} breaks the Assumption 3 with same advantage.*

Lemma 5. *Suppose that there is an adversary \mathcal{A} distinguish the $Game_{final,0}$ and $Game_{final,1}$ with a non-negligible advantage ϵ , then there is a simulator \mathcal{B} breaks the Assumption 4 with same advantage.*

We omitted the proof of Lemmas 4 and 5 because the space limitation.

Theorem 2. *If the Assumptions 1, 2, 3 and 4 hold, then our scheme is λ -leakage-resilient and anonymous for $\lambda \leq (\omega - 1 - 2c) \log p_2$, where c is a fixed positive constant.*

Proof. If these four assumptions hold, then from the Lemmas 1, 2, 3, 4 and 5, our scheme is λ -leakage-resilient and anonymous for $\lambda \leq (\omega - 1 - 2c) \log p_2$, where c is a fixed positive constant.

4.3 Leakage Performance

In this part, we give a concrete analysis of leakage resilience. The scheme has the same leakage bound $\lambda \leq (\omega - 1 - 2c) \log p_2$ with schemes [20–22] and the allowable probability $negl = p_2^{-c}$. Thus, the leakage rate of our scheme is $\gamma = \frac{\omega-1-2c}{(1+c_1+c_3)(\omega+1+3|S|)}$, where p_i ($i \in [4]$) is large primes of $d_i = c_i \kappa$ bits respectively. c_i is a positive constant.

4.4 Anonymity Analysis

To achieve the anonymity, we add the random elements in \mathbb{G}_{p_4} to components of public keys and the ciphertexts which has no effect on the decryption process because orthogonality. Next, we will give a concrete process to explain how to achieve anonymity.

$$\begin{aligned} \hat{e}(c_{x,1}, a_0) &= \hat{e}(T_{\rho(x)}^{\lambda_x} \cdot w_{\lambda_x,2}, g_1 \cdot w_0) \\ &= \hat{e}(g_1, g_1)^{t_{\rho(x)} \lambda_x} \hat{e}(w_{\rho(x)} w_{\lambda_x,2}, w_0)^{\lambda_x} \end{aligned} \quad (7)$$

$$\begin{aligned} \hat{e}(c_{x,0}, T_{i,j}) &= \hat{e}(a_0^{\lambda_x} \cdot w_{\lambda_x,1}, g_1^{t_{i,j}} \cdot w_{i,j}) \\ &= \hat{e}(g_1, g_1)^{t_{i,j} \lambda_x} \hat{e}(w_0 w_{\lambda_x,1}, w_{i,j})^{\lambda_x} \end{aligned} \quad (8)$$

In this case, we cannot decide the attribute value $\rho(x)$ in the access policy from the DDH-test even if $v_{i,j} = \rho(x)$, where $v_{i,j}$ is the attribute value for testing.

5 Performance Analysis

In this section, we will give a detailed analysis of the different schemes in terms of performance and efficiency in Tables 2 and 3, respectively.

As shown in Table 2, we compare these schemes [9, 22, 25, 27] with our construction in terms of revoicability, leakage-resilient and anonymity. [9, 22, 25] can support revocation, but all of them are not anonymous. In addition, [9] is not leakage-resilient. [27] cannot support revocation. However, our construction can achieve these three goals simultaneously.

Table 2. Performance comparisons among different ABE schemes

2 Scheme	Support revocation	Leakage-resilient	Anonymous
[9]	✓	×	×
[22]	✓	✓	×
[25]	✓	✓	×
[27]	×	✓	✓
Ours	✓	✓	✓

Let $\|\mathbb{G}\|, \|\mathbb{G}_T\|$ represent the size of the group \mathbb{G} and \mathbb{G}_T respectively. n is the number of attributes in universe attribute set, $|S|$ is the number of attributes in an attribute list S , l is the number of rows in \mathbf{A} , n' is the maximum number of users in the system. ω is the leakage parameter and P is the time of pairing operation.

Table 3. Efficiency comparisons among different ABE schemes

Scheme	Public parameter size	Private key size	Ciphertext size	Decryption time
[9]	$(2n + 2)\ \mathbb{G}\ $	$(2 + S)\ \mathbb{G}\ $	$(1 + 2l)\ \mathbb{G}\ + \ \mathbb{G}_T\ $	$(1 + 2 S)P$
[22]	$(\omega + n + 2n')\ \mathbb{G}\ + \ \mathbb{G}_T\ $	$(\omega + 2 S)\ \mathbb{G}\ $	$(\omega + 5l)\ \mathbb{G}\ + \ \mathbb{G}_T\ $	$(\omega + 4) \mathcal{R} P$
[25]	$(\omega + 3 + n)\ \mathbb{G}\ + \mathbb{G}_T\ $	$(\omega + 2 + S)\ \mathbb{G}\ $	$(\omega + 1 + 2l)\ \mathbb{G}\ + \ \mathbb{G}_T\ $	$(\omega + 1 + 2 S)P$
[27]	$(\omega + 3 + n)\ \mathbb{G}\ + \ \mathbb{G}_T\ $	$(\omega + 2 + S)\ \mathbb{G}\ $	$(\omega + 1 + 2l)\ \mathbb{G}\ + \ \mathbb{G}_T\ $	$(\omega + 1 + 2 S)P$
Ours	$(\omega + 4 + nn_i)\ \mathbb{G}\ + \ \mathbb{G}_T\ $	$(\omega + 1 + 3 S)\ \mathbb{G}\ $	$\ \mathbb{G}_T\ + (\omega + 1 + l + \sum_{x=1}^l l_x)\ \mathbb{G}\ $	$(\omega + 1 + \sum_{x=1}^{ \mathcal{R} } (2l_x + 2))P$

6 Conclusions

In this paper, a leakage-resilient CP-ABE scheme is proposed, which supports direct revocation and achieves adaptive security under four static assumptions in the standard model. Additionally, we show the proposed scheme achieves the anonymity based on the dual system encryption and composite order group. The performance analyses confirm the feasibility of our scheme. However, the proposed scheme relies on the composite order group, which issues a higher computation cost than a scheme in a prime order group under the same security standard. Designing a scheme with the same properties which is based on prime order bilinear group will be our future work.

References

1. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 457–473. ACM, Aarhus (2005)
2. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334. IEEE, Berkeley (2007)
3. Goyal, V., Pandey, O., Sahai, A., et al.: Attribute-based encryption for fine grained access control of encrypted data. In: 13th ACM Conference on Computer and Communications Security, pp. 89–98. ACM, Alexandria (2006)
4. Yu, S., Wang, C., Ren, K., et al.: Attribute based data sharing with attribute revocation. In: 5th ACM Symposium on Information, Computer and Communications Security, pp. 261–270. ACM, Beijing (2010)

5. Liang, X., Lu, R., Lin, X., et al.: Ciphertext policy attribute based encryption with efficient revocation (2010)
6. Hur, J., Noh, D.K.: Attribute-based access control with efficient revocation in data outsourcing systems. *IEEE Trans. Parallel Distrib. Syst.* **22**(7), 1214–1221 (2011)
7. Zhang, Y., Chen, X., Li, J., et al.: FDR-ABE: attribute-Based encryption with flexible and direct revocation. In: 5th International Conference on Intelligent Networking and Collaborative Systems, pp. 38–45. IEEE, Xi'an (2013)
8. Xie, X., Ma, H., Li, J., et al.: An efficient ciphertext-policy attribute-based access control towards revocation in cloud computing. *J. UCS* **19**(16), 2349–2367 (2013)
9. Naruse, T., Mohri, M., Shiraishi, Y.: Provably secure attribute-based encryption with attribute revocation and grant function using proxy re-encryption and attribute key for updating. *Hum.-Centric Comput. Inf. Sci.* **5**(1), 1–13 (2015)
10. Kapadia, A., Tsang, P., Smith, S.M.: Attribute-based publishing with hidden credentials and hidden policies. *NDSS* **7**, 179–192 (2007)
11. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: 27th Annual International Conference on Advances in Cryptology, pp. 146–162. ACM, Istanbul (2008)
12. Nishide, T., Yoneyama, K., Ohta, K.: Attribute-based encryption with partially hidden encryptor-specified access structures. In: 6th International Conference on Applied Cryptography and Network Security, pp. 111–129. ACM, New York (2008)
13. Li, J., Ren, K., Zhu, B., et al.: Privacy-aware attribute-based encryption with user accountability. In: 12th International Conference on Information Security, pp. 347–362. ACM, Pisa (2009)
14. Han, F., Qin, J., Zhao, H., et al.: A general transformation from KP-ABE to searchable encryption. *Future Gener. Comput. Syst.* **30**(1), 107–115 (2014)
15. Zhang, Y., Chen, X., Li, J., et al.: Anonymous attribute-based encryption supporting efficient decryption test. In: 8th ACM SIGSAC symposium on Information, Computer and Communications Security, pp. 511–516. ACM, Hangzhou (2013)
16. Chaudhari, P., Das, M.L., Mathuria, A.: On anonymous attribute based encryption. In: Jajodia, S., Mazumdar, C. (eds.) *ICISS 2015*. LNCS, vol. 9478, pp. 378–392. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26961-0_23
17. Zhang, Y., Zheng, D.: Anonymous attribute-based encryption with large universe and threshold access structures. In: *IEEE International Conference on Computational Science and Engineering and IEEE International Conference on Embedded and Ubiquitous Computing*, pp. 870–874. IEEE, Guangzhou (2017)
18. Zhang, L., Cui, Y., Mu, Y.: Improving privacy-preserving CP-ABE with hidden access policy. In: Sun, X., Pan, Z., Bertino, E. (eds.) *ICCCS 2018*. LNCS, vol. 11065, pp. 596–605. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00012-7_54
19. Zhang, L., Hu, G., Mu, Y., et al.: Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system. *IEEE Access* **7**, 33202–33213 (2019)
20. Lewko, A., Rouselakis, Y., Waters, B.: Achieving leakage resilience through dual system encryption. In: 8th Conference on Theory of Cryptography, pp. 70–88. ACM (2011)
21. Zhang, M., Shi, W., Wang, C., Chen, Z., Mu, Y.: Leakage-resilient attribute-based encryption with fast decryption: models, analysis and constructions. In: Deng, R.H., Feng, T. (eds.) *ISPEC 2013*. LNCS, vol. 7863, pp. 75–90. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38033-4_6

22. Zhang, M.: New model and construction of ABE: achieving key resilient-leakage and attribute direct-revocation. In: Susilo, W., Mu, Y. (eds.) ACISP 2014. LNCS, vol. 8544, pp. 192–208. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08344-5_13
23. Wang, Z., Yiu, S.M.: Attribute-based encryption resilient to auxiliary input. In: Au, M.-H., Miyaji, A. (eds.) ProvSec 2015. LNCS, vol. 9451, pp. 371–390. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26059-4_21
24. Zhang, L., Zhang, J., Hu, Y.: Attribute-based encryption resilient to continual auxiliary leakage with constant size ciphertexts. *J. China Univ. Posts Telecommun.* **23**(3), 18–28 (2016)
25. Yu, Q., Li, J.: Continuous leakage resilient ciphertext-policy attribute-based encryption supporting attribute revocation. *Comput. Eng. Appl.* **52**(20), 29–38 (2016)
26. Zhang, L., Zhang, J., Mu, Y.: Novel leakage-resilient attribute-based encryption from hash proof system. *Comput. J.* **60**(4), 541–554 (2017)
27. Zhang, J., Zhang, L.: Anonymous CP-ABE against side-channel attacks in cloud computing. *J. Inf. Sci. Eng.* **33**(3), 789–805 (2017)



Cryptographic Reverse Firewalls for Identity-Based Encryption

Yuyang Zhou¹, Yuanfeng Guan², Zhiwei Zhang², and Fagen Li¹(✉)

¹ University of Electronic Science and Technology of China, Chengdu 611731, China
fagenli@uestc.edu.cn

² SI-TECH Information Technology Co., Ltd, Beijing, China

Abstract. The Snowden revelations show that powerful attackers can compromise user's machines to steal users' private information. At the same time, many of the encryption schemes that are proven to be secure in Random Oracle Model (ROM) may present undetectable vulnerabilities when implemented, and these vulnerabilities may reveal a users' secrets, e.g., the machine hides some backdoors without the user's awareness, and an attacker can steal the user's private information through these backdoors. Recently, Mironov and Stephens-Davidowitz proposed cryptographic reverse firewall (CRF) to solve this problem. However, there is no CRF for identity-based encryption (IBE) has been proposed. In this paper, we propose two CRF protocols for IBE. One is a one-round encryption protocol with CRF used on the receiver, and the other is a two-round encryption protocol with CRFs deployed on both sender and receiver. We prove that these two protocols can resist the exfiltration of secret information and one is only secure against a chosen plaintext attack (CPA), the other is semantically secure against an adaptive chosen ciphertext attack (IND-ID-CCA). Moreover, we use JPBC to implement our protocols. The experimental results indicate that our protocols have some advantages in communication cost. Under certain computation cost conditions, our protocols are efficient and practical.

Keywords: Identity-based encryption · Cryptographic reverse firewalls · Exfiltration resistance

1 Introduction

For a long time, ordinary people think if they use a series of security policies promulgated by the National Security Agency, then the message sent by them to their friends online will be completely confidential during the transmission process. However, the revelations of Edward Snowden show that the US National Security Agency monitors their every online action through subverting cryptographic standards [2], intercepting and tampering with hardware on its way to users [3]. Ironically, the US government have always claimed that they would not interfere with the privacy of every legal citizen. Meanwhile, some serious security flaws were found in cryptographic modules. These vulnerabilities can easily

cause serious attacks on the system, and user’s information was exposed to the attacker [4–6]. All of these have led to the emergence of a new research direction, which named post-Snowden cryptography [7]. The main problems solved by the post-Snowden cryptography can be summarised as: “When an attacker can arbitrarily interfere with a user’s computer, such as secretly installing a backdoor on the user’s computer, and how can the security of the user’s information be guaranteed?” This a very interesting question.

Just a simple cryptographic protocol, such as symmetric-key encryption and public-key encryption protocols can no longer meet such requirements. To solve this problem, we have been motivated by the cryptographic reverse firewall (CRF) [8]. CRF can be seen as a further study of the “black-box” cryptography of the last century [9,10]. It provides a common framework for how to maintain the security of the transmitted information on a compromised computer. CRF can be seen as a protocol that “sits between” the user and the outside world. This protocol modifies the message sent and received by the user. In other words, the CRF can confuse the attacker whether the message he intercepted is the user’s real message he needs. In [8], Mironov and Stephens-Davidowitz use CRF to construct an oblivious transfer protocol. For the IBE scheme, there also may have the backdoor exploit. So can we use CRF to construct an identity-based encryption (IBE) protocol to solve the problems faced by the post-Snowden cryptography?

1.1 Related Work

Traditional public key cryptosystem uses a trusted third party called a certificate authority (CA) to maintain the public keys of all users. Although CA can guarantee the authenticity of the public key, it increases the system’s computational cost. Meanwhile, with the increase of the user’s public key, certificate management becomes more and more complicated. To get rid of CA management and simplify key management, Shamir [16] proposed identity-based cryptosystem (IBC) in 1984. Its main idea is that the user’s public key is a binary string, which is calculated from the user’s personal information (e.g., user’s name, phone number, e-mail address, etc.). Therefore, this binary string is uniquely identified. Currently, some effective IBE [11–13] schemes are proposed. However, there is a fatal weakness in IBC, the key escrow problem. For example, the public key generator (PKG) can easily impersonate any user, it can decrypt any ciphertext in the IBE scheme and is not easy to be founded.

In 2015, Mironov and Stephens-Devidowitz [8] first proposed the concept of the cryptographic reverse firewall (CRF) that can provide strong security in the presence of active insider attackers, such as backdoors. Meanwhile, they designed an oblivious transfer protocol that had a secure CRF for each user. Finally, they provided a generic structure to protect users from data leakage and eavesdropping through any protocol with CRF. Next year, Dodis, Mironov and Stephens-Devidowitz [17] considered the message-transmission protocols in the CRF framework. They presented a rich set of solutions that vary in different

setup assumptions, security and efficiency. Surprisingly, they verified their solutions could achieve CCA security and CPA security against adversaries. Chen et al. [18] introduced the notion of the malleable smooth projective hash function (SPHF) and how to generically construct CRFs using malleable SPHFs in a modular way for some widely used cryptographic protocols. Recently, Ma et al. [19] proposed a concessive online/offline ciphertext-policy attribute-based encryption with CRFs, and they verified their scheme could achieve exfiltration resistance for users and selective CPA security. However, the above mentioned cryptographic schemes constructed with CRFs do not involve a specific identity-based cryptographic protocol.

1.2 Motivation and Contribution

Most IBE schemes are not resistant exfiltration of secret information attacks. To achieve exfiltration-resistant, we design the following two cryptographic reverse firewalls (CRFs), which based on the Boneh and Franklin’s IBE scheme [12].

1. We construct a one-round encryption protocol with CRF deployed on the receiver. The receiver’s CRF performs key malleability operation on the public keys and then sends the public keys to the sender who wants to encrypt messages. When receiver’s CRF receive the ciphertext which encrypted by these public keys, it will perform a restore key malleability operation. So for the PKG, even it knows the receiver’s private key, as long as the receiver’s CRF is trusted, the user’s information will not eavesdrop.
2. We construct a two-round encryption protocol with CRFs deployed on both sender and receiver. The receiver’s CRF performs key malleability operation on the public keys and sender’s CRF performs re-randomisation encryption operation on message. Under the premise of maintaining functionality, and security the original security of the BF’s full IBE scheme, our scheme is also resistant to the exfiltration attacks.

1.3 Organization

The rest of the paper is arranged as follows. The definitions of IBE and CRF are introduced in Sect. 2. A one-round identity-based encryption protocol with CRF and its CPA security analysis are showed in Sect. 3. A two-round identity-based encryption protocol with CRFs and its IND-ID-CCA security analysis are showed in Sect. 4. The performance of these two CRFs’ schemes is analysed in Sect. 5. Finally, the conclusion is given in Sect. 6.

2 Preliminaries

In this section, we briefly review some definitions of identity-based encryption. Since CRF is a relatively new concept, we give a specific introduction to framework and properties of the CRF.

2.1 Bilinear Pairing and Identity-Based Encryption

Definition 1 (Bilinear Pairing). Let \mathbb{G}_1 and \mathbb{G}_2 be a addition and a multiplication group of prime order q respectively. If there is a bilinear mapping $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and meets the following three features, we can say \hat{e} is a bilinear pairing. Where, P is the generator of group \mathbb{G}_1 and three features are:

- (1) *Bilinearity:* For any $P, Q \in \mathbb{G}_1$, $a, b \in \mathbb{Z}_q^*$, we have $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
- (2) *Non-degeneracy:* Let $P, Q \in \mathbb{G}_1$ and $1_{\mathbb{G}_2}$ be the unit, we have $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$.
- (3) *Computability:* For any $P, Q \in \mathbb{G}_1$, there is a valid algorithm that can compute $\hat{e}(P, Q)$.

Definition 2 (Identity-based Encryption (IBE)). An IBE scheme consists of four parts: **Setup**, **Extract**, **Encrypt**, and **Decrypt**.

- **Setup:** A PKG performs probabilistic polynomial time (PPT) operation in the case of input security parameter k , outputs and exposes the system's public parameter par , and keeps the master key msk secret.
- **Extract:** A PKG inputs par , msk and a user's identity $ID \in \{0, 1\}^*$, and performs key extraction operation to output the user's private key S_{ID} .
- **Encrypt:** A sender enters par , a plaintext message m and a receiver's identity ID , performs a PPT encryption operation, and outputs ciphertext c to the receiver with ID .
- **Decrypt:** After entering par , c , ID and S_{ID} the receiver with ID performs a deterministic decryption operation, outputs m or the error symbol " \perp ".

It is important to note that this algorithm should be consistent. That is, if $c = \mathbf{Encrypt}(par, m, ID)$, it must satisfy $m = \mathbf{Decrypt}(par, c, ID, S_{ID})$.

Definition 3 (Chosen Ciphertext Security (IND-CCA)). Chosen ciphertext security (IND-CCA) is the standard acceptable notion of security for a public key encryption scheme. IBE schemes also need to ensure IND-CCA security. However, IBE schemes use the user's identity, the definition of IND-CCA needs to be extended. We say that an IBE scheme is semantically secure against an adaptive chosen ciphertext attack (IND-ID-CCA) if no polynomially bounded adversary \mathcal{A} has a non-negligible advantage against the Challenger \mathcal{C} in the following game:

- **Initial:** \mathcal{C} runs the **Setup** algorithm with a security parameter k and sends the public parameters par to \mathcal{A} .
- **Phase 1:** \mathcal{A} can perform a polynomially bounded number of queries in an adaptive manner, which means each query may depend on the responses to the previous queries.
 1. *Key extraction queries:* \mathcal{A} chooses an identity ID . \mathcal{C} calculates the private key $S_{ID} = \mathbf{Extract}(par, ID, msk)$ and sends it to \mathcal{A} .
 2. *Decryption queries:* \mathcal{A} chooses an identity ID and a ciphertext c . \mathcal{C} computes $S_{ID} = \mathbf{Extract}(par, ID, msk)$ and $c = \mathbf{Decrypt}(par, ID, c, S_{ID})$, then outputs c to \mathcal{A} . Note that this result may be error if c is invalid ciphertext, then \mathcal{C} will output \perp to \mathcal{A} .

- **Challenge:** \mathcal{A} decides when **Phase 1** ends. \mathcal{A} generates two equal length plaintexts, m_0 and m_1 , and an identity ID^* , on which it wants to be challenged. Note that it cannot be asked for S_{ID^*} in **Phase 1**. \mathcal{C} takes a random bit γ from $0, 1$ and computes $c^* = \mathbf{Encrypt}(par, m_\lambda, ID^*)$, then sends c^* to \mathcal{A} .
- **Phase 2:** \mathcal{A} can ask a polynomial bounded number of queries adaptively again as in **Phase 1** with the limit that it cannot make a key extraction query on ID^* and cannot make a decryption query on (c^*, ID^*) to obtain the corresponding plaintext.
- **Guess:** \mathcal{A} outputs a bit γ' and wins the game if $\gamma' = \gamma$.

We refer to such an adversary \mathcal{A} as an IND-ID-CCA attacker and define \mathcal{A} 's advantage in attacking the scheme as: $\text{Adv}(\mathcal{A}) = |\Pr[\gamma' = \gamma] - \frac{1}{2}|$.

One point that needs to be added is that if there is no decryption query in this game, then the scheme can achieve chosen plaintext attack (CPA) security, which is one wayness (OW) secure.

2.2 Cryptographic Reverse Firewall

A CRF can be interpreted as a machine that sits between the user's computer and the outside world, and messages that pass through CRF can be modified. Below we summarise definition and some properties of CRF from [8]. We strongly recommend the reader those interested in CRF read [8] for more detailed discussions.

Definition 4 (Cryptographic Reverse Firewall (CRF)). *A cryptographic reverse firewall (CRF) is a stateful algorithm \mathcal{W} that takes as input its state and a message, and outputs an updated state and message. For simplicity, we do not write the state of \mathcal{W} explicitly. For a CRF \mathcal{W} and a party $P=(\text{receive}, \text{next}, \text{output})$, the composed party is defined as*

$$\begin{aligned} \mathcal{W} \circ P &:= (\text{receive}_{\mathcal{W} \circ P}(\sigma, m) = \text{receive}_P(\sigma, \mathcal{W}(m)), \\ &\quad \text{next}_{\mathcal{W} \circ P}(\sigma) = \mathcal{W}(\text{next}_P(\sigma)), \\ &\quad \text{output}_{\mathcal{W} \circ P}(\sigma) = \text{output}_P(\sigma)). \end{aligned}$$

When the composed party engages in a protocol, the state of \mathcal{W} is initialized to the public parameters ρ . If \mathcal{W} is meant to be composed with a party P , we call it a CRF for P .

A qualified CRF is required to maintain the functionality of the underlying protocol, preserve the same security as the properly implemented protocol, and prevent the machine from leaking any information to the outside world.

Definition 5 (Functionality-maintaining CRFs). *For any CRF \mathcal{W} and any party P , let $\mathcal{W} \circ P = \mathcal{W} \circ (\mathcal{W}^{k-1} \circ P)$. For any protocol \mathcal{P} that satisfies some functionality requirements function \mathcal{F} , we say that if $\mathcal{W}^k \circ P_j$ maintains \mathcal{F} for P_j in \mathcal{P} for any polynomially bounded $k \geq 1$, then the \mathcal{W} maintains \mathcal{F} for P_j in \mathcal{P} . When \mathcal{F} , P_j and \mathcal{P} are clear, we simply say that \mathcal{W} maintains functionality.*

Definition 6 (Security-preserving CRFs). For a protocol \mathcal{P} that satisfies some security requirements \mathcal{S} and functionality \mathcal{F} and a CRF \mathcal{W} .

- (1) If the protocol $\mathcal{P}_{P_j \Rightarrow \mathcal{W} \circ P_A^*}$ satisfies \mathcal{S} for any PPT P_A^* , then \mathcal{W} strongly preserves \mathcal{S} for P_j in \mathcal{P} ; and
- (2) If the protocol $\mathcal{P}_{P_j \Rightarrow \mathcal{W} \circ P_A^*}$ satisfies \mathcal{S} for any PPT P_A^* that maintains \mathcal{F} , then \mathcal{W} weakly preserves \mathcal{S} for P_j in \mathcal{P} against \mathcal{F} -maintaining adversaries.

Definition 7 (Exfiltration-resistant CRFs). For a protocol \mathcal{P} that satisfies some security requirements \mathcal{S} and functionality \mathcal{F} and a CRF \mathcal{W} .

- (1) If no PPT adversary \mathcal{A} achieves advantage that is non-negligible in the security parameter λ in the game $\mathbf{LEAK}(\mathcal{P}, P_j, J, \mathcal{W}, \lambda)$, then \mathcal{W} is (\mathcal{P}, P_j, J) -strongly exfiltration-resistant; and
- (2) If no PPT adversary \mathcal{A} achieves advantage that is non-negligible in the security parameter λ in the game $\mathbf{LEAK}(\mathcal{P}, P_j, J, \mathcal{W}, \lambda)$ provided that the \mathcal{A} 's output P_A^* maintains \mathcal{F} for P_j , then \mathcal{W} is (\mathcal{P}, P_j, J) -weakly exfiltration-resistant against \mathcal{F} -maintaining adversaries.

When \mathcal{F} , P_j and \mathcal{P} are clear, we simply say that \mathcal{W} is strongly exfiltration-resistant against \mathcal{J} or weakly exfiltration-resistant against \mathcal{J} respectively. When \mathcal{J} is empty, we can say that \mathcal{W} is exfiltration-resistant eavesdroppers.

3 A One-Round IBE Protocol with CRF

In 2001, Boneh and Franklin proposed two IBE schemes. The first one is a basic IBE scheme which is only secure against a chosen plaintext attack (CPA). The second one is their full IBE scheme, which extends the basic scheme to get against an adaptive chosen ciphertext attack (IND-ID-CCA) in the random oracle model. We design two different CRFs for these two IBE schemes. In this section, we first introduce a one-round IBE protocol with CRF which based on the Boneh and Franklin's basic IBE scheme [12].

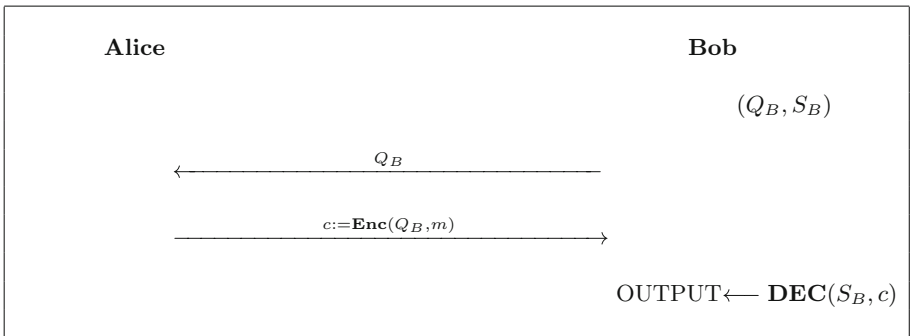


Fig. 1. A basic IBE scheme

Let Alice want to send an encrypted message to Bob. A basic IBE scheme is that Bob send his public key Q_B to Alice, and Alice uses Q_B to encrypt plaintext m , then send ciphertext c to Bob. Figure 1 shows this simple process. Note that a complete IBE includes two parts: system setup and users' keys extraction. Since our schemes with CRFs have not changed much in these two parts, we will not mark it separately in figures.

We construct a one-round encryption protocol with CRF deployed only on Bob's side. To provide a CRF for Bob, our scheme must be key malleable (KeyMaul), which can be seen in Fig. 2. The most important step in KeyMaul is the operation on (Q_B, P_{pub}) . This process can be seen as a process of rerandomization of keys. In our scheme, we add two parts (i.e., **KeyMaul** and **ReKeyMaul**) to the original four parts (i.e., **Setup**, **Extract**, **Encrypt**, and **Decrypt**). A one-round IBE scheme with CRF is described below.

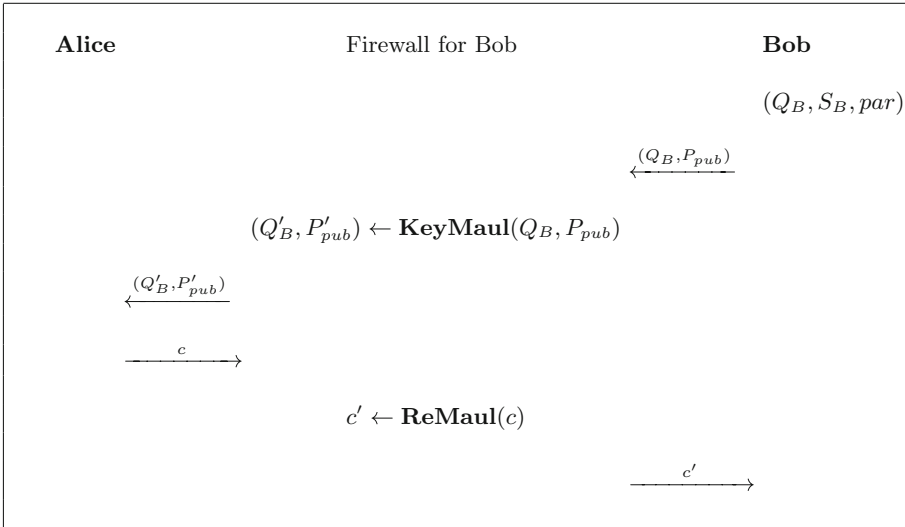


Fig. 2. A one-round IBE scheme with Bob's CRF

- **Setup:** Given a security parameter k , PKG selects a additive group \mathbb{G}_1 and a multiplicative group \mathbb{G}_2 of prime order q , a bilinear pair $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, two hash functions, which are $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \rightarrow \{0, 1\}^n$. n is the number bits of the encrypted message. PKG selects a master key $s \in \mathbb{Z}_q^*$ and keeps it secret, then calculates own public key $P_{pub} = sP$, P is the generator of \mathbb{G}_1 . Finally, PKG public system public parameter $par = \{\mathbb{G}_1, \mathbb{G}_2, q, n, \hat{e}, P, P_{pub}, H_1, H_2\}$.
- **Extract:** Bob submits his ID_B to PKG, PKG calculates Bob's public key $Q_B = H_1(ID_B)$ and Bob's private key $S_B = sQ_B$ and sends (Q_B, S_B, par) to Bob online or offline. If the online transmission is used, we can use a secure socket layer (SSL) protocol to ensure the keys' confidential.

- **KeyMaul**: Since Alice wants to send an encrypted message to Bob, Bob passes his public key to Alice. During the transmission of Q_B to Alice, Q_B first passes through Bob’s CRF. Bob’s CRF can modify (Q_B, P_{pub}) randomly, here we call this process KeyMaul. It is a randomized algorithm that inputs (Q_B, P_{pub}) and outputs $(Q'_B, P'_{pub}) \leftarrow (\alpha Q_B, \beta P_{pub})$ that be sent to Alice, where $(\alpha, \beta) \in (\mathbb{Z}_q^*)^2$ are chosen uniformly and independently randomly.
- **Encrypt**: Alice selects a random number $r \in \mathbb{Z}_q^*$, and then computes $g_B \leftarrow \hat{e}(Q'_B, P'_{pub})$, $U = rP$, $V = m \oplus H_2(g_B^r)$. Finally, Alice sends ciphertext $c = (U, V)$ to Bob.
- **ReKeyMaul**: When Bob’s CRF receives $c = (U, V)$, it will perform a restore **KeyMaul** operation, that is compute $U' \leftarrow \alpha\beta U$.
- **Decrypt**: When Bob receives $c = (U', V)$, he will decrypt c by computing $m = V \oplus H_2(\hat{e}(S_B, U'))$ and verify the correctness of m . If $U' \in \mathbb{G}_1^*$ of prime order q holds, Bob accepts m as a valid message; otherwise, Bob rejects m and outputs the error symbol “ \perp ”.

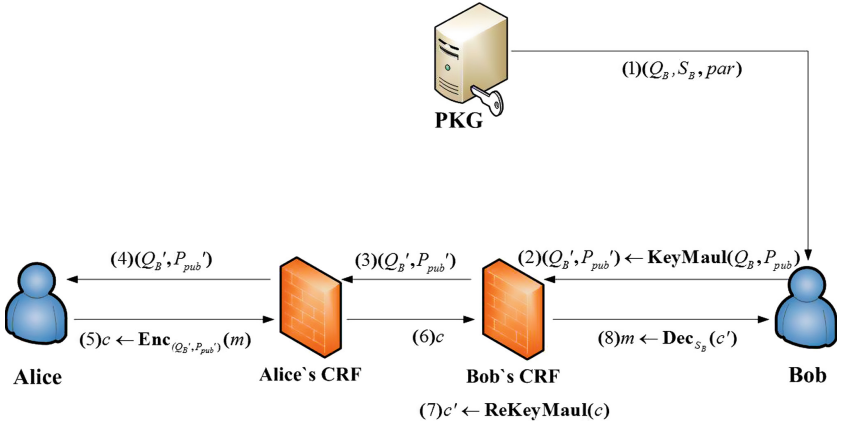


Fig. 3. System model of a one-round encryption protocol with CRF

Figure 3 summarises the complete process of a one-round encryption protocol with CRF. From the Fig. 3 we can see that PKG does not know the user’s actual public key for encrypting the message, and PKG can’t use user’s private key to decrypt ciphertext c for PKG doesn’t know the random variables (α, β) generated by user’s CRF. Therefore, our proposed scheme uses CRF to prevent backdoor leaks similar to those that occurred in Snowden revelations. However,

Theorem 1. *The proposed one-round encryption protocol with CRF is one-way identity-based encryption scheme (ID-OWE), and the CRF for Bob maintains functionality, weakly preserve security, and weakly resist exfiltration if the Boneh and Franklin’s basic IBE scheme [12] can achieve ID-OWE security.*

Proof. We show that our construction satisfies the following properties.

- *Functionality Maintaining:* The correctness can be easily verified. When Alice encrypts the message m , it will computes

$$\begin{aligned} H_2(g_B^r) &= H_2(\hat{e}(Q'_B, P'_{pub})^r) \\ &= H_2(\hat{e}(\alpha Q_B, \beta P_{pub})^r) \\ &= H_2(\hat{e}(\alpha Q_B, \beta sP)^r) \\ &= H_2(\hat{e}(sr\alpha\beta Q_B, P)). \end{aligned}$$

When Bob's CRF receives c , it will convert c by

$$\begin{aligned} H_2(\hat{e}(S_B, \alpha\beta U)) &= H_2(\hat{e}((\alpha\beta sQ_B), rP)) \\ &= H_2(\hat{e}(sr\alpha\beta Q_B, P)). \end{aligned}$$

Therefore,

$$H_2(\hat{e}(S_B, \alpha\beta U)) = H_2(g_B^r).$$

- *Weak Security Preservation and Weak Exfiltration Resistance:* Because ID-OWE is a very weak security requirement, the only ciphertext cannot be restored to its plaintext, which is the basic requirement of the encryption algorithm. At the same time, because the safety proof of our proposed scheme 2-a two-round IBE protocol with CRFs (two-round-IBE-CRFs) is similar to that of this one-round-IBE-CRF scheme, we will introduce the specific security analysis in Sect. 4. For those interested in the BF's basic IBE scheme's ID-OWE security analysis, please read paper [12], this excellent paper will inspire you.

4 A Two-Round IBE Protocol with CRFs

We construct a two-round encryption protocol with CRFs deployed on both sides, which based on the Boneh and Franklin's full IBE scheme [12]. To provide a CRF for Bob, our scheme must be key malleable (KeyMaul), and this process is similar to KeyMaul's introduction in the Sect. 3. To provide a CRF for Alice, the encryption part in our scheme must be re-randomizable (Rerand), which can be seen in Fig. 4. The most important step in Rerand is the operation of g_B . In our scheme, we add three parts (i.e., **KeyMaul**, **ReEncrypt** and **ReDecrypt**) to the original four parts (i.e., **Setup**, **Extract**, **Encrypt**, and **Decrypt**). A two-round IBE scheme with CRFs is described below.

- **Setup:** Given a security parameter k , PKG selects a additive group \mathbb{G}_1 and a multiplicative group \mathbb{G}_2 of prime order q , a bilinear pair $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$, four hash functions, which are $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2 \rightarrow \{0, 1\}^*$, $H_3 : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \mathbb{Z}_q^*$ and $H_4 : \{0, 1\}^n \rightarrow \{0, 1\}^n$. n is the number bits of the encrypted message. PKG selects a master key $s \in \mathbb{Z}_q^*$ and keeps it secret, then calculates own public key $P_{pub} = sP$, P is the generator of \mathbb{G}_1 . Finally, PKG public system public parameter $par = \{\mathbb{G}_1, \mathbb{G}_2, q, n, \hat{e}, P, P_{pub}, H_1, H_2, H_3, H_4\}$.

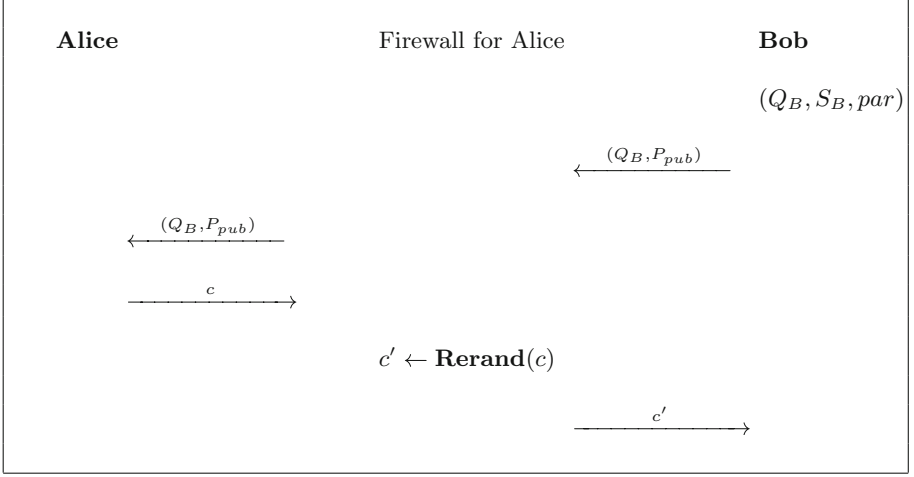


Fig. 4. A two-round IBE scheme with Alice's CRF

- **Extract:** Same as the **Extract** part of a one-round IBE protocol with CRF in Sect. 3.
- **KeyMaul:** Since Alice wants to send an encrypted message to Bob, Bob passes his public key to Alice. During the transmission of Q_B to Alice, Q_B first passes through Bob's CRF. Bob's CRF can modify (Q_B, P_{pub}) randomly, here we call this process KeyMaul. It is a randomized algorithm that inputs (Q_B, P_{pub}) and outputs $(Q'_B, P'_{pub}) \leftarrow (\alpha Q_B, \beta P_{pub})$ that be sent to Alice, where $(\alpha, \beta) \in (\mathbb{Z}_q^*)^2$ are chosen uniformly and independently randomly.
- **Encrpt:** When Alice'CRF receives (Q'_B, P'_{pub}) , it will carry out a randomization encryption operation. Specifically, Alice's CRF chooses $\varepsilon \in \mathbb{Z}_q^*$ randomly, and then computes $(Q'_B, P'_{pub}) \leftarrow (Q'_B + \varepsilon P, P'_{pub})$, then these variables to Alice.
- **ReEncrypt:** Alice selects a random number $\theta \in \{0, 1\}^n$ in advance, and then computes $r = H_3(\theta, m)$, $g_B \leftarrow \hat{e}(Q'_B, P'_{pub}) \cdot \hat{e}(\varepsilon P, P_{pub})$, $U = rP$, $V = \theta \oplus H_2(g_B^r)$, $W = m \oplus H_4(\theta)$. Finally, Alice wants to send ciphertext $c = (U, V, W)$ to Bob.
- **Decrypt:** When Bob's CRF receives c , it will perform a restore **KeyMaul** operation, that is compute $U' \leftarrow \alpha\beta U$.
- **ReDecrypt:** When Bob receives $c = (U, V, W, U')$, he will compute $\theta' = V \oplus H_2(\hat{e}(S_B, U'))$. Next, Bob decrypts c by computing $m = W \oplus H_4(\theta')$ and $r' = H_3(\theta', m)$. Finally, Bob verifies the correctness of m . If $U = r'P$ holds, Bob accepts m as a valid message; otherwise, Bob rejects m and outputs the error symbol “ \perp ”.

What we want to explain is that some readers may think that the fourth step **Encrpt** is also the process of **keyMaul**. In fact, the process of **keyMaul** also utilizes a re-randomization. Here, the key extension is indeed an encryption

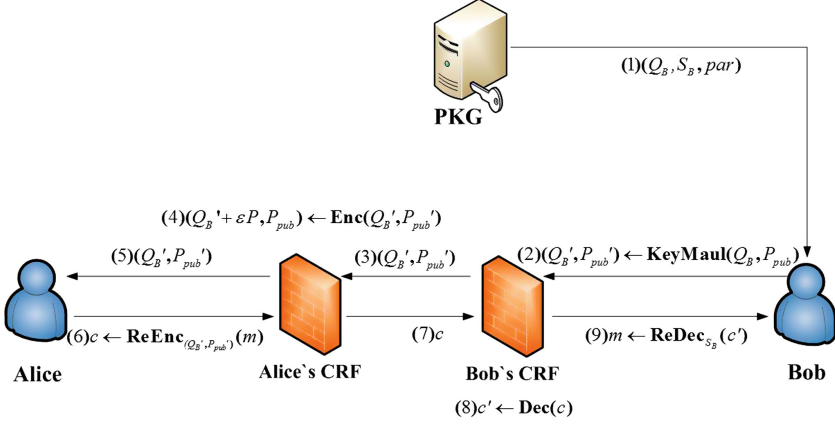


Fig. 5. System model of a two-round encryption protocol with CRFs

process for the encryptor Alice. As for why we only consider Q_B and P_{pub} , because these two variables are related to the generation of $g_B \leftarrow g_B \cdot \hat{e}(\varepsilon P, P_{pub})$. Where, $g_B = \hat{e}(Q_B, P_{pub})$ is originally calculated by Alice using Q_B and P_{pub} . Figure 5 summarises the complete process of a two-round encryption protocol with CRFs. From the Fig. 5 we can see that PKG does not know the user's actual public key for encrypting the message, and PKG can't use user's private key to decrypt ciphertext c for PKG doesn't know the random variables (α, β) generated by user's CRFs. Therefore, our proposed scheme uses CRFs to prevent backdoor leaks similar to those that occurred in Snowden revelations.

Theorem 2. *The proposed two-round encryption protocol with CRFs is IND-ID-CCA, and the CRFs for Bob and Alice maintain functionality, weakly pre-preserve security, and weakly resist exfiltration if the Boneh and Franklin's full IBE scheme [12] can achieve IND-ID-CCA security.*

Proof. We show that our construction satisfies the following properties.

- *Functionality Maintaining:* The correctness can be easily verified. When Alice's CRF performs the operation of rerandomizable encryption, it will select $\varepsilon \in \mathbb{Z}_q^*$ randomly and then compute

$$\begin{aligned}
 H_2(g_B^r) &= H_2(\hat{e}(Q'_B + \varepsilon P, P'_{pub})^r) \\
 &= H_2(\hat{e}(\alpha Q_B, \beta P_{pub})^r \hat{e}(\varepsilon P, \beta P_{pub})^r) \\
 &= H_2(\hat{e}(\alpha Q_B, \beta s P)^r \hat{e}(P, P)^{\varepsilon \beta s r}) \\
 &= H_2(\hat{e}(sr\alpha\beta Q_B, P)).
 \end{aligned}$$

When Bob's CRF receives c , it will convert c by

$$\begin{aligned}
 H_2(\hat{e}(S_B, \alpha\beta U)) &= H_2(\hat{e}(\alpha\beta s Q_B, r P)) \\
 &= H_2(\hat{e}(sr\alpha\beta Q_B, P)).
 \end{aligned}$$

Therefore,

$$H_2(\hat{e}(S_B, \alpha\beta U) = H_2(g_B^r).$$

- *Weak Security Preservation and Weak Exfiltration Resistance:* We prove the IND-ID-CCA security of our proposed schemes with tampered algorithms by proving the indistinguishability between the security game of our two-round-IBE-CRFs and the BF’s full IBE [12]. First, we briefly introduce the IND-ID-CCA-CRF game between our two-round-IBE-CRFs and the BF’s full IBE [12]. It is similar to the security game in Sect. 2, except that for decryption queries, challenger \mathcal{C} runs **Decrypt** and **ReDecrypt**. Next, we consider the following games:

- **Game 1.** It is identical to the security game of IND-ID-CCA-CRF just above said.
- **Game 2.** Same as **Game 1** except that during the *Phase 1* and *Phase 2*, the conversion public keys (Q_B, P_{pub}) are generated by the **Extract** algorithm, not the **KeyMaul** algorithm.
- **Game 3.** Same as **Game 2** except that during the *Challenge* phase, the challenge ciphertext c^* is generated by the **Extract** algorithm in BF’s full IBE, not the **Encrypt** and **ReEncrypt** algorithm in our proposed two-round IBE protocol with CRFs. In fact, **Game 3** is the security game of the BF’s full IBE scheme.

Then we prove the indistinguishability between the pairs **Game 1** and **Game 2**, **Game 2** and **Game 3** respectively. For the pair **Game 1** and **Game 2**, for any tampered algorithm **Setup** and **Extract**, after the post-processing by the reviewer’s CRF \mathcal{W}_B , the (Q_B, P_{pub}) are uniformly random due to the key malleability, which is identical to the original algorithm **Setup**, thus **Game 1** and **Game 2** are indistinguishable. Since the pair **Game 2** and **Game 3**, for any tampered algorithm **Encrypt** and **ReEncrypt**, after the post-processing by the sender’s CRF \mathcal{W}_A , the updated ciphertext c are uniformly regenerated because the IBE scheme is rerandomization, which is identical to the encryption algorithm in the BF’s full IBE scheme, thus **Game 2** and **Game 3** are indistinguishable. Therefore, we conclude that **Game 1** and **Game 3** are indistinguishable. Since the BF’s full IBE scheme can achieve IND-IN-CCA security, our proposed two-round-IBE-CRFs can also achieve IND-IN-CCA security.

The IND-IN-CCA security of the proposed scheme indicates that the CRFs for data sender and data receiver, maintain weakly preserve security. The indistinguishability between **Game 1** and **Game 3** indicates that the CRFs for data sender and data receiver, maintain weakly resist exfiltration. Combining all the discussions, we complete the proof.

5 Performance Analysis

In this section, we discuss these two schemes’ performance. First, we theoretically analysed the communication cost of the schemes and then used the JPBC library to implement the schemes and analyse the computational cost of the schemes.

5.1 Theoretical Analysis

Currently, there is no scheme for applying CRF to IBE. So we compare our design with the BF's basic and full IBE schemes [12], a concessive online/offline attribute-based encryption with cryptographic reverse firewall (COO-CP-ABE-CRF) [19]. Because the addition operation, exponent operation, pairing operation, and point multiplication operation are the most expensive in the whole scheme, and other operations are negligible compared with them. So we use these four operations as a measure to calculate the basic operation of cost. Table 1 shows the comparison of the calculation costs and communication costs of these schemes. Here, we use PM to denote the point multiplication operation in the group G_1 , use Exp to denote the exponent operation in the group G_2 , and use P to denote the pairing operation on the bilinear map. For communication costs, we use $|m|$ to denote the number of bits of message m , $|G_1|$ indicates the number of bits of an element in group G_1 , $|G_2|$ indicates the number of bits of an element in group G_2 , $|\mathbb{Z}_q^*|$ indicates the number of bits of an element in the group \mathbb{Z}_q^* , $|ID|$ indicates the number of bits of the group ID . At the same time, because of the attributed-based scheme, we use l to indicate the length of the attribute set involved, n_s to represent the number of group members in the self-organising network, and $|S|$ to represent the number of bits in the attribute organisation.

From Table 1, we can see that attribute-based cryptosystem universal storage makes the private key length too long. It also has the characteristics of the computation cost, the length of the attribute set and the number of group members in the self-organising network are linearly increased. Therefore, the computation cost in the scheme [19] cannot obtain a value in this paper. It can only be known that the scheme [19] does not have any advantage in terms of computation cost and communication cost. Although the computational cost of our schemes is more expensive than BF's IBE schemes [12], our schemes have better security to resist the exfiltration of secret information. At the same time, the computational cost of our schemes is within the system's ability to withstand. Therefore, our schemes are better than the BF's original schemes in practical application.

Table 1. Comparison of schemes performance

Scheme	Computation cost		Communication cost
	Sender	Receiver	
[19]	$(3l + n_s + 3)\text{Exp}$	$(l + n_s)\text{Exp} + (3 + 2l)\text{P}$	$(2l + 3) G_1 + G_2 + S $
[12]1	PM+Exp+P	P	$ G_1 + m $
[12]2	PM+Exp+P	PM+P	$ G_1 + 2 m $
ours1	PM+Exp+P	3PM+P	$ G_1 + m $
ours2	2PM+Exp+P+Add	4PM+P	$ G_1 + 2 m $

In order to more intuitively draw the advantages of our scheme in communication cost, we set $|m|=160$ bits, $|G_1| = 513$ bits, and $|G_2| = 1024$ bits.

Figure 6 shows the comparison of the communication costs of these schemes. For scheme [19], we set the length of the attribute set l in 0 bit, and the number of bits $|S|$ of the elements in the attribute mechanism is 0 bit. In reality, l and $|S|$ cannot be 0 bits. However, in this paper, we only want to highlight the minimum communication cost in the scheme [19], and our scheme also dominates the communication cost. As can be seen from Table 1 and Fig. 6, our scheme has certain advantages in communication cost and security in theoretical analysis.

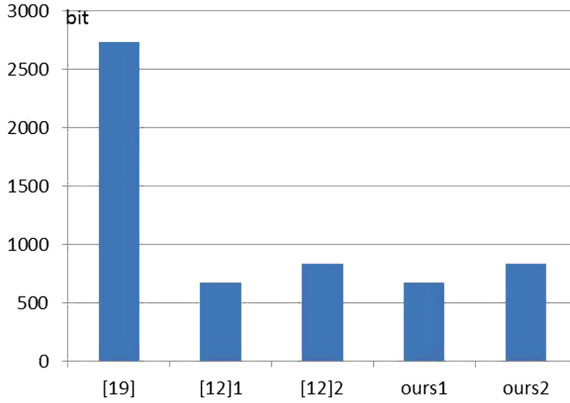


Fig. 6. Comparison in communication cost

5.2 Experimental Analysis

In this subsection, we implemented our schemes using the JPBC library. Constructing a bilinear pair here, we use asymmetric pairing based on the elliptic curve $y^2 = x^3 + x \pmod q$ in the finite field $\mathbb{E}(\mathbb{F}_p)$. Considering the security of the protocol, we take $p=512$ bits, the order q of the cyclic group is a large prime number of 160 bits. So the output of H_3 is 160 bits. Since G_1 is a cyclic addition group on the finite field $\mathbb{E}(\mathbb{F}_p)$, P is the generator of G_1 , so the size of P is 1024 bits. The size of P_{pub} is 1024 bits, and the output of the Hash function H_1 is also 1024 bits. Here we use the secure Hash function SHA-256, so the H_2 and H_4 outputs are both 256 bits.

The experimental development environment is Eclipse, Neon.1a Release (4.6.1). The computer configuration of the program execution environment is Intel(R) Core(TM) i5-5200U CPU @ 2.20 GHz 2.19 GHz processor, 8 GB of RAM, 64-bit Windows operating system. To make the experimental values more representative, we cycle through the entire steps of the access control scheme 1000 times to get the average time taken to complete each algorithm. Experiment shows that our first one-round IBE scheme with CRF's average running time is 244 ms, which is setup time is 29 ms, extract time is 73 ms, keyMaul time is 45 ms, encrypt time is 37 ms, reKeyMaul time is 44 ms, decrypt time is 13 ms and Fig. 8 shows that the computing cost of the firewall in a one-round CRF

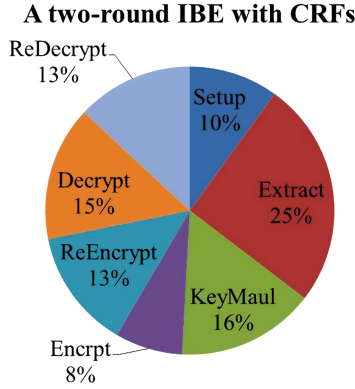


Fig. 7. Ration of each phase running time in our two-round scheme

scheme is about 31% of the total scheme. Experiment shows that our second two-round IBE scheme with CRFs' average running time is 295 ms, which is setup time is 29 ms, extract time is 74 ms, keyMaul time is 45 ms, encrypt time is 22 ms, reEncrypt time is 39 ms, decrypt time is 44 ms, reDecrypt time is 38 ms and Fig. 7 shows that the computing cost of the firewall in a two-round CRFs scheme is about 39% of the total scheme. What we can know is that install CRFs in a traditional IBE scheme will use the system's more computation cost. However, compared with BF's IBE schemes, our schemes can resist the exfiltration of secret information and have better security. So in the case of acceptable computing costs, our protocols have great advantages than BF's IBE schemes.

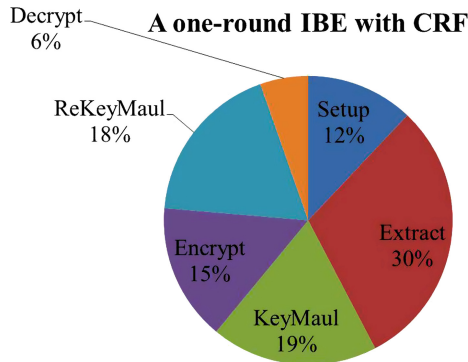


Fig. 8. Ration of each phase running time in our one-round scheme

6 Conclusion

In this paper, we proposed two identity-based encryption with cryptographic reverse firewalls, which can resist the exfiltration of secret information. Furthermore, compared with the attribute-based encryption with cryptographic reverse firewalls, our protocols have a great advantage in computation and communication costs. Compared with the Boneh and Franklin's IBE schemes, there is a certain weakness in computation cost for our protocols. But it is within acceptable limits. Next, we plan to improve our protocols further, reduce its computational cost, and find a general-purpose framework for configuring cryptographic reverse firewalls for identity-based encryption schemes.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (grant no. 61872058).

References

1. Fang, X., Misra, S., Xue, G., Yang, D.: Smart grid-the new and improved power grid: a survey. *IEEE Commun. Surv. Tutorials* **14**(4), 944–980 (2011)
2. Perlroth, N., Larson, J., Shane, S.: N.S.A. Able to Foil Basic Safeguards of Privacy on Web. *The New York Times*, New York (2013)
3. Greenwald, G.: No Place to Hide: Edward Snowden, the NSA, and the U.S. Surveillance State. Metropolitan Books, New York (2014)
4. Vulnerability summary for CVE-2014-1260('Heartbleed'), April 2014. <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-1260>
5. Vulnerability summary for CVE-2014-1266 ('goto fail'), February 2014. <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-1266>
6. Vulnerability summary for CVE-2014-6271('Shellshock'), September 2014. <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-6271>
7. Tang, D.Q.: Cliptography: post-snowden cryptography. In: Proceedings of the ACM SIGSAC Conference on Computer & Communications Security 2017, pp. 2615–2616. ACM, Dallas, TX, USA (2017)
8. Mironov, I., Stephens-Davidowitz, N.: Cryptographic reverse firewalls. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 657–686. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_22
9. Young, A., Yung, M.: The dark side of "Black-Box" cryptography or: should we trust capstone? In: Kobitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 89–103. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68697-5_8
10. 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>
11. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45325-3_32
12. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. *SIAM J. Comput.* **32**(3), 586–615 (2003)
13. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (Without Random Oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_17

14. Hess, F.: Efficient identity based signature schemes based on pairings. In: Nyberg, K., Heys, H. (eds.) SAC 2002. LNCS, vol. 2595, pp. 310–324. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36492-7_20
15. Choon, J.C., Hee Cheon, J.: An identity-based signature from gap diffie-hellman groups. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 18–30. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36288-6_2
16. Shamir, A.: Indentity-based crytosystems and signature schemes. LNCS **21**(2), 47–53 (1984)
17. Dodis, Y., Mironov, I., Stephens-Davidowitz, N.: Message transmission with reverse firewalls—secure communication on corrupted machines. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 341–372. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_13
18. Chen, R., Mu, Y., Yang, G., Susilo, W., Guo, F., Zhang, M.: Cryptographic reverse firewall via malleable smooth projective hash functions. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10031, pp. 844–876. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_31
19. Ma, H., Zhang, R., Yang, G., Song, Z., Sun, S., Xiao, Y.: Concessive online/offline attribute based encryption with cryptographic reverse firewalls—secure and efficient fine-grained access control on corrupted machines. In: Lopez, J., Zhou, J., Soriano, M. (eds.) ESORICS 2018. LNCS, vol. 11099, pp. 507–526. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98989-1_25



Identity-Based Encryption Resilient to Continual Leakage Without Random Oracles

Yuyan Guo, Mingming Jiang^(✉), Shimin Wei, Ming Xie,
and Mei Sun

School of Computer Science and Technology, Huaibei Normal University,
Huaibei 235000, Anhui, China
jiangmm3806586@126.com

Abstract. In general, the security of identity-based encryption schemes has been considered under the ideal circumstances, where the adversaries do not acquire the secret internal state of the schemes. However, the adversaries can obtain partial information for the secret key through the various key leakage attacks in reality. In order to further describe the continual leakage attack, we formally define a secure model for identity-based encryption. The adversary is allowed to continuously acquire part of the secret information through the continual leakage attack in the secure model. Then we give a new type identity-based encryption scheme resilient to continual leakage. This scheme which is based on an identity-based key encapsulation mechanism is secure against chosen-ciphertext attack under the hardness of the computational bilinear Diffie-Hellman problem in the standard model. This proposed scheme enhances the continual leakage-resilient property and enjoys less computation cost.

Keywords: Identity-based encryption · Continual leakage · key encapsulation mechanism

1 Introduction

To address the problem of certificate management in the traditional public key cryptosystem, Shamir proposed the idea of identity-based encryption (IBE) [1] in 1984, where a user's identity is its public key and the corresponding secret key is generated by the Private Key Generator (PKG). Then, Boneh and Franklin [2] constructed the first efficient IBE scheme from the weil pairing in 2001. Canetti et al. [3] put forward an IBE scheme which is chosen-identity security in the standard model. Waters [4] in 2005 and Gentry [5] in 2006 respectively brought forward an IBE scheme that is fully secure in the standard model. In recent years, many secure and efficient IBE scheme [6–9] have been proposed.

In the real world, most cryptographic schemes which are proved secure in an ideal model are not able to resist key leakage caused by the side channel attacks. Leakage-resilient cryptography can capture the side channel attacks by modeling information leakage that adversary can access. To formalize side channel attacks, the cryptographic

researchers began to study the leakage models which typically have: only computation leaks information model [10], relative-leakage model [11], bounded-retrieval model (BRM) [12], auxiliary inputs model [13], continual leakage model [14] and after-the-fact leakage model [15]. Especially in the continual leakage model, the adversaries can continue to acquire a bounded amount of secret internal state information for the cryptographic primitive. Provably secure cryptography schemes in the presence of the key leakage have attracted a lot of attention recently. One of the most important research direction in the field is to design leakage-resilient IBE schemes. Galindo et al. [16] provided a master-key leakage-resilient IBE against master-key leakage attacks. Sun et al. [17] constructed a practical leakage-resilient fully adaptively chosen ciphertext attacks (CCA) secure IBE scheme in the standard model, and its leakage parameter is independent of the message length. Li et al. [18] applied a hash proof technique to construct a new leakage-resilient IBE scheme in the BRM, which is more computationally efficient than the Alwen et al.'s leakage-resilient IBE [19] scheme. Yuen et al. [20] defined an after-the-fact auxiliary input model, and provided an IBE scheme with after-the-fact auxiliary inputs. Specifically for the continual leakage model, Li et al. [21] proposed the formal definition and security model of identity-based broadcast encryption with continual leakage-resilience. Based on the dual system encryption technique, the scheme is proved to be secure under subgroup decisional assumptions. Then, Li et al. [22] gave an IBE scheme under composite order groups, which is secure against after-the-fact continuous auxiliary input in the standard model. Zhou et al. [23] gave a new CCA-secure IBE scheme tolerating continual leakage in the standard model, and its security is proved in the selective-ID security model based on the hardness of decisional bilinear Diffie-Hellman assumption. Then, Zhou et al. [24] constructed a continuous leakage-resilient CCA-secure IBE scheme with leakage amplification which is proved secure in the standard model. The benefit of their scheme [21–24] is that the length of permitted leakage can be adjusted flexibly according to the continual leakage requirements.

Motivations and Contributions. There have been many IBE schemes proposed in this context, almost all of which, however, can only achieve chosen plaintext attack (CPA) security. We propose the outline and security model of IBE resilient to continual leakage. Then, we put forward an IBE scheme which is provably CCA secure in the standard model and is able to resist the continual leakage. Our IBE scheme is based on an identity-based key encapsulation mechanism. We randomize the encapsulated symmetric key using the strong extractor, and the encapsulated symmetric key which is allowed to be leaked is used to encrypt the message. We provide a secret key update algorithm to tolerate the continual leakage.

Paper Organization. In Sect. 2, we review some preliminaries that are used in the paper. In Sect. 3, we give the outline and security model for IBE resilient to continual leakage. In Sect. 4, we construct an IBE scheme resilient to continual leakage. We prove that our IBE scheme is CCA secure in Sect. 5. Efficiency comparison is shown in Sect. 6. Finally, we put forward the conclusion in Sect. 7.

2 Preliminaries

Definition 1. Let \mathbb{G} and \mathbb{G}_T be two multiplicative cyclic groups of the same order p . Let g be a generator of \mathbb{G} . e is a bilinear map if $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ has the properties as follows:

- Bilinear: $e(P_1^a, P_2^b) = e(P_1, P_2)^{ab}$ for all $P_1, P_2 \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$.
- Non-degenerate: $e(g, g) \neq 1_{\mathbb{G}_T}$.
- Computable: The map e is efficiently computable.

The security of our IBE scheme resilient to continual leakage depends on the following computational bilinear Diffie-Hellman (CBDH) difficult problem.

Definition 2. Let \mathbb{G} and \mathbb{G}_T be two multiplicative cyclic groups of the same order p . Let g be a generator of \mathbb{G} . e is a bilinear map. We define the CBDH problem: Given $\mathcal{D} = (g, A, B, C) \in \mathbb{G}^4$, where $A = g^a, B = g^b, C = g^c, a, b, c \in \mathbb{Z}_p^*$, an adversary A computes $T = e(g, g)^{abc}$.

The advantage that a probabilistic polynomial time (PPT) adversary A solves the CBDH problem is defined as $\text{Adv}_A^{\text{CBDH}} = \Pr[A(\mathcal{D}) = e(g, g)^{abc}]$. We say that the CBDH problem is hard if $\text{Adv}_A^{\text{CBDH}}$ is negligible for all PPT adversaries.

Combined with Definition 2 and the Goldreich-Levin theorem [25], we have following lemma in the bilinear setting for a Goldreich-Levin hardcore predicate $f: \mathbb{G}_T \times \{0,1\}^\mu \rightarrow \{0,1\}$, where $\mu \in \mathbb{N}$.

Lemma 1. Let $A, B, C \in \mathbb{G}$, $u \in \{0,1\}^\mu$, $K = f(T, u)$, and let $K' \in \{0,1\}$ be uniformly random. Suppose there exists a PPT algorithm B distinguishing the distributions $\Delta = (\mathcal{D}, K, u)$ and $\Delta_{\text{rand}} = (\mathcal{D}, K', u)$ with non-negligible advantage. Then there exists a PPT algorithm computing T on input $\mathcal{D} = (g, A, B, C)$ with non-negligible advantage, hence breaking the CBDH problem.

Definition 3. The min-entropy of a random variable X is: $H_\infty(X) = -\log(\max_x \Pr[X = x])$.

Definition 4. For random variable X, Y , the average conditional min-entropy is defined as $\tilde{H}_\infty(X|Y) = -\log\left(E_{y \leftarrow Y} \left[\max_x \Pr[X = x | Y = y] \right]\right) = -\log\left(E_{y \leftarrow Y} \left[2^{-H_\infty(X|Y=y)} \right]\right)$ where $E_{y \leftarrow Y}$ denotes the expected value over all values of the random variable Y .

Lemma 2. For any random variables X, Y, Z such that Y has 2^l ($l \in \mathbb{N}$) possible values, we have that $\tilde{H}_\infty(X|(Y, Z)) \geq \tilde{H}_\infty(X|Z) - l$.

Definition 5. The statistical distance between two random variables X, Y is defined by $\text{SD}(X, Y) = \frac{1}{2} \sum_x |\Pr[X = x] - \Pr[Y = x]|$, where $x \in F$ and F is a finite domain.

Definition 6. An efficient randomized function $\text{Ext}: \{0,1\}^n \times \{0,1\}^\mu \rightarrow \{0,1\}^\eta$ is an average-case (m, ε) -strong extractor if for all X, Y such that $n, \mu, \eta \in \mathbb{N}$, $X \in \mathbb{G}$, Y has 2^l possible values, $m \in \mathbb{N}$ and $\tilde{H}_\infty(X|Y) \geq m$, ε is a negligible value. We get

$SD((Ext(X, U_\mu), U_\mu, Y), (U_\eta, U_\mu, Y)) \leq \varepsilon$, where \mathbb{G} is a nonempty set, and the two variables U_μ, U_η are uniformly distributed over $\{0, 1\}^\mu, \{0, 1\}^\eta$ respectively.

3 The Outline and Security Model of IBE Resilient to Continual Leakage

3.1 The Outline of IBE Resilient to Continual Leakage

Our IBE scheme has five algorithms (*Setup*, *KeyGen*, *Enc*, *Dec*, *UpdateSK*). We add an *UpdateSK* algorithm to update secret key, and the size of the corresponding updated secret key remains the same.

Setup. Given a security parameter 1^λ ($\lambda \in \mathbb{N}$), the algorithm generates a master public key mpk and a master secret key msk .

KeyGen. Given mpk, msk and an identity ID , the algorithm outputs a secret key sk_{ID} .

Enc. On input mpk, ID and a message M , the algorithm returns the ciphertext C .

Dec. Given sk_{ID} and C , the algorithm outputs M or \perp if C is an invalid ciphertext.

UpdateSK. Given sk_{ID} and mpk , the algorithm generates an updated secret key sk'_{ID} where $|sk'_{ID}| = |sk_{ID}|$.

3.2 Security Model for IBE Resilient to Continual Leakage

Referring to [4, 5, 26], we propose a formal definition of continual leakage model for IBE. The security for our IBE scheme against continual leakage, selective-identity and adaptively chosen ciphertext attacks is defined through the following game CL-sID-CCA. The challenger Ch creates a list L_{sk} to store the tuple in the form of (ID, sk_{ID}) , L_{sk} is empty in the initialization for the game.

Init. A sends the identity ID^* to Ch .

Setup. Ch first runs *Setup* algorithm, then outputs mpk to the adversary A .

Phase 1. The following oracles are inquired through A adaptively.

- Secret Key Oracle: Given $ID \neq ID^*$, Ch checks the tuple (ID, sk_{ID}) in L_{sk} . If it does not exist, Ch runs *KeyGen* algorithm, outputs a secret key sk_{ID} . (ID, sk_{ID}) is added to L_{sk} .
- Leakage Oracle: Ch creates a list L_{leak} containing the tuples like (ID, K, cnt) , K denotes the secret information which is used to encrypt the message, and $cnt \in \mathbb{N}$ is a counter. L_{leak} is empty at the beginning of the game. Ch checks (ID, K, cnt) from L_{leak} . If the tuple does not exist, Ch adds $(ID, K, 0)$ into L_{leak} . After the step or if the tuple exists, Ch determines if $cnt + l_i \leq l$ where $i \in \mathbb{N}$. If it's not true, it outputs \perp . Otherwise, it sets $cnt \leftarrow cnt + l_i$ for (ID, K, cnt) and outputs $f_i(K)$, where f_i is a leakage function and $f_i : \mathbb{G}_T \rightarrow \{0, 1\}^{l_i}$.

- Decryption Oracle: Given ID , Ch responds by running $KeyGen$ algorithm to generate the secret key sk_{ID} . Ch then runs Dec algorithm to decrypt the ciphertext C using sk_{ID} . Ch sends the resulting plaintext to A .

Challenge Phase. A submits M_0, M_1 with the same size to Ch . Ch randomly picks $\beta \in \{0, 1\}$ for encryption. It then returns $C^* = Enc(params, mpk, M_\beta, ID^*)$ to A .

Phase 2. A continues making the queries as in the Phase 1 with the following restriction: A is not allowed to issue decryption queries on (ID^*, C^*) .

Guess. A returns a guess $\beta' \in \{0, 1\}$. A wins this game if $\beta' = \beta$.

The advantage for A in an IBE scheme is $Adv_A^{CL-sID-CCA} = |\Pr[\beta = \beta'] - \frac{1}{2}|$.

Definition 7. We say that an IBE scheme is CL-sID-CCA secure in the continual leakage model, if the advantage $Adv_A^{CL-sID-CCA}$ of any PPT adversary A in the above game is negligible.

4 Our IBE Scheme Resilient to Continual Leakage

Referring to [26, 27], we give a new IBE scheme resilient to continual leakage in the standard model. We use the strong extractor technology and an identity-based key encapsulation mechanism to construct this scheme. Our scheme consists of the following five algorithms:

Setup. Let \mathbb{G} and \mathbb{G}_T be two cyclic groups of order p . g is the generators of \mathbb{G} . Let $l = l(n)$ be a bound of the total leakage. The algorithm picks $\alpha \in \mathbb{Z}_p^*$, $h, X', Y_1, \dots, Y_n \in \mathbb{G}$ randomly where $n \in \mathbb{N}$, and sets $X = g^\alpha$. The algorithm picks a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ and a strong extractor $Ext : \{0, 1\}^n \times \{0, 1\}^\mu \rightarrow \{0, 1\}^\eta$, $n, \mu, \eta \in \mathbb{N}$, defines two hash functions $H_1 : \mathbb{Z}_p^* \rightarrow \mathbb{G}$ as $ID \rightarrow X^{ID}h$, $H_2 : \mathbb{G} \rightarrow \mathbb{Z}_p^*$ and a function $f : \mathbb{G}_T \times \{0, 1\}^\mu \rightarrow \{0, 1\}$ where $\mu \in \mathbb{N}$. The master secret key $msk = \alpha$ and $mpk = \{g, \mathbb{G}, \mathbb{G}_T, e, Ext, f, H_1, H_2, h, X, X', Y_1, \dots, Y_n\}$.

KeyGen. For an identity ID , the algorithm randomly chooses $x_i \in \mathbb{Z}_p^*$ where $i \in [1, n]$. It returns $sk_{ID} = \{sk_i\}_{i \in [1, n]}$ where $sk_i = (sk_{i,1}, sk_{i,2}) = (Y_i^\alpha H_1(ID)^{x_i}, g^{x_i})$. It sends sk_{ID} to the user in the security channel.

Enc. For the message M , the algorithm randomly picks $r \in \mathbb{Z}_p^*$, $u \in \{0, 1\}^\mu$ where $\mu \in \mathbb{N}$. then computes $C_1 = g^r$, $C_2 = (X^\tau X')^r$ where $\tau = H_2(C_1)$, $C_3 = H_1(ID)^r$, $K = (K_1 \parallel \dots \parallel K_n)$ where $K_i = f(e(X, Y_i)^r, u)$ for $i \in [1, n]$, $C_4 = Ext(K, u) \oplus M$ and $C_5 = u$. The ciphertext is $C = \{C_1, C_2, C_3, C_4, C_5\}$.

Dec. Given ciphertext $C = \{C_1, C_2, C_3, C_4, C_5\}$ and a secret key $sk_{ID} = \{sk_i\}_{i \in [1, n]}$, the algorithm first computes $\tau = H_2(C_1)$. If $e(C_1, X^\tau X') \neq e(g, C_2)$ or $e(C_1, H_1(ID)) \neq e(g, C_3)$, it outputs \perp , else, it computes $K_i = f\left(\frac{e(C_1, sk_{i,1})}{e(C_3, sk_{i,2})}, C_5\right)$ and $K = (K_1 \parallel \dots \parallel K_n)$, and outputs $M = C_4 \oplus Ext(K, C_5)$.

For a valid ciphertext, we have $\frac{e(C_1, sk_{i,1})}{e(C_3, sk_{i,2})} = \frac{e(g^r, Y_i^2 H_1(ID)^{x_i})}{e(H_1(ID)^{x_i}, g^{x_i})} = e(X, Y_i)^r$.

UpdateSK. Given $sk_{ID} = \{sk_i\}_{i \in [1, n]}$ where $sk_i = (sk_{i,1}, sk_{i,2})$, the secret key update algorithm randomly chooses $x'_i \in \mathbb{Z}_p^*$ where $i \in [1, n]$. It outputs a new secret key $sk'_{ID} = \{sk'_i\}_{i \in [1, n]}$ where $sk'_i = (sk'_{i,1}, sk'_{i,2}) = (sk_{i,1} \cdot H_1(ID)^{x'_i}, sk_{i,2} \cdot g^{x'_i})$.

5 Security Analysis

Theorem 1. If the CBDH problem holds, our IBE scheme is CL-sID-CCA secure.

Proof. We prove the security by a sequence of games. Let $C^* = \{C_1^*, C_2^*, C_3^*, C_4^*, C_5^*\}$ denote the challenge ciphertext with the corresponding encapsulated symmetric key K^* of the identity ID^* , let K'^* denote the random value chosen in the CL-sID-CCA game. Let $\tau^* = H_2(C_1^*)$. We start with a game which is the same as the CL-sID-CCA game, and end up with a game where both K^* and K'^* are chosen randomly. All the games below are indistinguishable under the CBDH problem. Let E_i denote the event that the adversary A outputs β' such that $\beta' = \beta$ in the Game i .

Game 0. This game is the same as the CL-sID-CCA game. We have $\Pr[E_0] = \frac{1}{2} + Adv_A^{\text{CL-sID-CCA}}$.

Game 1. Let E_{01} be the event that A submits the decryption query $\langle C'_1, C'_2, C'_3, C'_4, C'_5 \rangle$ for ID^* with $C'_1 = C_1^*$ in the Phase 1. The probability that A issues the decryption query such that $C'_1 = C_1^*$ before seeing the challenge ciphertext is bounded by q_D/p , where q_D is the number of decryption inquiries. Since $q_D = \text{poly}(\lambda)$ where $\text{poly}(\cdot)$ is a polynomial function, we have $\Pr[E_{01}] \leq q_D/p \leq \text{negl}(\lambda)$ where $\text{negl}(\lambda)$ is a negligible value. Game 1 is similar to Game 0 except assuming that E_{01} never happens in Game 1. Thus we have that $|\Pr[E_0] - \Pr[E_1]| \leq \text{negl}(\lambda)$, and we know that if $C'_1 = C_1^*$, the decryption query $\langle C'_1, C'_2, C'_3, C'_4, C'_5 \rangle$ for ID^* is not allowed in Phase 2.

Game 2. Let E_{12} be the event that A submits the decryption query $\langle C'_1, C'_2, C'_3, C'_4, C'_5 \rangle$ for ID^* with $C'_1 \neq C_1^*$ and $H_2(C'_1) = H_2(C_1^*)$. Due to the collision resistance of the hash function H_2 , we have $\Pr[E_{12}] \leq \text{negl}(\lambda)$. Game 2 is similar to Game 1 except assuming that E_{12} never happens in Game 2. Thus we have that $|\Pr[E_1] - \Pr[E_2]| \leq \text{negl}(\lambda)$.

Game 3. Game 3 is similar to Game 2 except assuming that $K^* \in \{0,1\}^{nv}$ is chosen randomly, where $n, v \in \mathbb{N}$. K'^* is a uniformly random, both K^* and K'^* are picked randomly, thus we have $\Pr[E_3] = 1/2$.

We conclude that $|\Pr[E_2] - \Pr[E_3]| \leq \text{negl}(\lambda)$ under the CBDH problem. The results will be proved by a hybrid argument. We first define a sequence of games Game(0), ..., Game(n), such that Game(0) is the same as Game 2 and Game(n) is the same as Game 3. We argue that Game(i) is indistinguishable from Game($i-1$) based on the CBDH problem, where $i \in [1, n]$. Since we have that Game(0) is the same as Game 2.

Then, for i from 1 to n , the first iv bits for K^* are set to be random in $\text{Game}(i)$, and the rest is the same as in $\text{Game}(i - 1)$. Thus, $\text{Game}(n)$ is the same as Game 3 . Let W_i be the event that A outputs β' such that $\beta' = \beta$ in $\text{Game}(i)$. Suppose that $|\Pr[W_0] - \Pr[W_n]| = 1/\text{poly}'(\lambda)$ where $\text{poly}'(\cdot)$ is a polynomial function. In other word, the advantage of A in $\text{Game}(0)$ which is close to the advantage in $\text{Game}(n)$ is not ignored. There must exist a subscript i such that $|\Pr[W_{i-1}] - \Pr[W_i]| = 1/\text{poly}(\lambda)$.

Suppose $|\Pr[W_0] - \Pr[W_n]| = 1/\text{poly}'(\lambda)$ holds, we can construct an algorithm B distinguishing the distributions between Δ and Δ_{rand} in Lemma 1 for the CBDH problem. B takes a challenge $A = (\mathcal{D}, R, u)$ as input, where R is either a random value from $\{0, 1\}^v$ or v bits formed by $f(T, u)$, guesses the subscript $j \in [1, n]$ with the probability at least $1/n$ such that $|\Pr[W_{j-1}] - \Pr[W_j]| = 1/\text{poly}(\lambda)$, and interacts with A as follows.

Init. A sends the challenging identity ID^* .

Setup. B first chooses $d \in \mathbb{Z}_p^*$, sets $X = A = g^a$, $X' = X^{-\tau^*} g^d$, $Y_j = B = g^b$ where $\tau^* = H_2(C)$. For $i \in [n] \setminus \{j\}$, B chooses $y_i \in \mathbb{Z}_p^*$ and computes $Y_i = g^{y_i}$, then selects $z \in \mathbb{Z}_p^*$ and sets $h = X^{-ID^*} g^z$. B sends to A $mpk = \{g, \mathbb{G}, \mathbb{G}_T, e, \text{Ext}, f, H_1, H_2, h, X, X', Y_1, \dots, Y_n\}$. $msk = a$ is unknown to B . We define the function H_1 as the form $H_1(x) = X^x h = X^{x-ID^*} g^z$.

Phase 1. The following oracles are inquired through A adaptively.

- Secret Key Oracle: Given $ID \neq ID^*$, B checks the tuple (ID, sk_{ID}) in L_{sk} . If it does not exist, B generates the secret key $sk_{ID} = \{sk_j\}_{j \in [1, n]}$: B randomly chooses $x_j \in \mathbb{Z}_p^*$ and sets $sk_j = (sk_{j,1}, sk_{j,2}) = \left(Y_j^{\frac{-x_j}{ID-ID^*}} H_1(ID)^{x_j}, g^{x_j} Y_j^{\frac{-1}{ID-ID^*}} \right)$. For sk_i where $i \in [n] \setminus \{j\}$, B randomly chooses $x_i \in \mathbb{Z}_p^*$ and sets $(sk_{i,1}, sk_{i,2}) = (X^{y_i} H_1(ID)^{x_i}, g^{x_i}) = (Y_i^a H_1(ID)^{x_i}, g^{x_i})$. Let $\tilde{x}_j = x_j - \frac{b}{(ID-ID^*)}$. Due to $sk_{j,1} = Y_j^{\frac{-x_j}{ID-ID^*}} H_1(ID)^{x_j} = Y_j^a (X^{ID-ID^*} g^z)^{x_j - \frac{b}{(ID-ID^*)}} = Y_j^a H_1(ID)^{\tilde{x}_j}$ and $sk_{j,2} = g^{x_j} Y_j^{\frac{-1}{ID-ID^*}} = g^{\tilde{x}_j}$, we know that sk_{ID} is a valid secret key of ID . (ID, sk_{ID}) is added to L_{sk} .
- Leakage Oracle: B creates a list L_{leak} containing the tuples like (ID, K, cnt) , K denotes the secret information which is used to encrypt the message, and $cnt \in \mathbb{N}$ is a counter. L_{leak} is empty at the beginning of the game. B checks (ID, K, cnt) from L_{leak} . If the tuple does not exist, B adds $(ID, K, 0)$ into L_{leak} . After the step or if the tuple exists, B determines if $cnt + l_i \leq l$ where $i \in \mathbb{N}$. If it's not true, it outputs \perp . Otherwise, it sets $cnt \leftarrow cnt + l_i$ for (ID, K, cnt) and outputs $f_i(K)$, where f_i is a leakage function and $f_i: \mathbb{G}_T \rightarrow \{0, 1\}^{l_i}$.
- Decryption Oracle: Given $\langle ID, C_1, C_2, C_3, C_4, C_5 \rangle$, B responds as follows. If $ID \neq ID^*$, B responds by running KeyGen algorithm to generate the secret key sk_{ID} , then runs Dec algorithm to decrypt the ciphertext using sk_{ID} . Otherwise, B computes $\tau = H_2(C_1)$ and checks the consistency of the ciphertext by verifying $e(C_1, X^\tau X') \stackrel{?}{=} e(g, C_2) \wedge e(C_1, H_1(ID)) \stackrel{?}{=} e(g, C_3)$, if it is true, B sets $K = (K_1 \parallel \dots \parallel K_n)$ where $K_i = f(e(X, C_1)^{y_i}, u)$, $i \in [n] \setminus \{j\}$, and $K_j = f(e(\tilde{X}, Y_j), u)$

where $\tilde{X} = (C_2/C_1^d)^{1/(\tau-\tau^*)} = (X^{r(\tau-\tau^*)}g^{rd}/g^{rd})^{1/(\tau-\tau^*)} = X^r$. Through Game 2, we have that when $ID = ID^*$, if $C_1 \neq C_1^*$, we have $\tau \neq \tau^*$. B computes $M = C_4 \oplus \text{Ext}(K, C_5)$. B returns the decryption queries and sends the plaintext to A .

Challenge Phase. A submits M_0, M_1 with the same size and a target identity ID^* to B . B randomly picks $u^* \in \{0, 1\}^\mu$ and sets $C_1^* = C$ (which implies $r = c$) where $C = g^c$, $C_2^* = C^d$, $C_3^* = C^z$. B randomly picks $i - 1$ groups of v bits $K_{1,1}^*, \dots, K_{1,j-1}^*$, sets $K_{1,j}^* = R$ (where R is either a random value from $\{0, 1\}^v$ or v bits formed by $f(T, u)$) and $K_{1,i}^* = f(e(X, C_1^*)^{j_i}, u^*)$ ($i \in [j+1, n]$), $K_1^* = (K_{1,1}^* \parallel \dots \parallel K_{1,n}^*)$. B randomly picks $\beta \in \{0, 1\}$ and $K_0^* \in \{0, 1\}^{nv}$, sets $C_{M_\beta}^* = \text{Ext}(K_\beta^*, u^*) \oplus M_\beta$ and $C_5^* = u^*$. Since $(X^{\tau^*} X')^r = (g^d)^r = C^d$ and $H_1(ID^*)^r = (g^z)^r = C^z$, the challenge ciphertext $C_\beta^* = (C_1^*, C_2^*, C_3^*, C_{M_\beta}^*, C_5^*)$ is valid. B outputs the challenge ciphertext to A .

Phase 2. A continues making the queries as in the Phase 1 with the following restriction: A is not allowed to issue decryption queries on (ID^*, C^*) .

Guess. A returns a guess $\beta' \in \{0, 1\}$. If $\beta' = \beta$, B outputs 1 which means $K_1^* \in \Delta$, A 's view is identical to $\text{Game}(i - 1)$. Otherwise, B outputs 0 which means $K_0^* \in \Delta_{\text{rand}}$, A 's view is identical to $\text{Game}(i)$. Thus B is able to distinguish the distributions Δ and Δ_{rand} . According to the Lemma 1, there exists a PPT algorithm which can break the CBDH problem. However, the CBDH problem is hard, there is a contradiction.

Leakage Ratio Analysis

For the leakage analysis of symmetric key $K \in \{0, 1\}^n$, let set Z includes public parameter and secret key. The adversary A gains no more than l bits leakage for K . According to the Lemma 2, we know that $\tilde{H}_\infty(K|L, Z) \geq \tilde{H}_\infty(K|Z) - l = n - l$, where L is a random variable with l bits length. We choose the $(n - l, \epsilon)$ -strong extractor. One of the ciphertext $C_4 = \text{Ext}(K, u) \oplus M$ and the uniform distribution is not distinguishable, $n - l$ can be close to zero, and the leakage bound l is approximately equal to n , the leakage ratio of K is $l/n \approx n/n = 1$.

6 Efficiency Comparison

We compare schemes [14, 26], Π and Π_{New} in [23] with our scheme on security properties and performance. The details are listed in Tables 1 and 2.

From Table 1, five schemes are proved secure in the standard model, and possesses the security property of continual leakage.

Let \mathbb{G}_T be a cyclic groups of order $N = p_1 p_2 p_3$ where p_1, p_2, p_3 are distinct primes. Let $\mathbb{G}_{p_1}, \mathbb{G}_{p_3}$ be the subgroups of order p_1, p_3 in \mathbb{G} respectively. m, n are integers. $|\mathbb{Z}_q|$ is the size of element in \mathbb{Z}_q . We analyze the efficiency of the five schemes as follows.

From Table 2, the lengths of the master public key, secret key, and so on for our scheme are smaller than [14, 26]. Thus, the performance of our scheme is slightly better

Table 1. Security properties comparison

Scheme	Model	Hard problem	Continual leakage
[14]	Standard	Three static assumption in composite order bilinear groups	√
[26]	Standard	Three static assumption in composite order bilinear groups	√
[23] - \prod	Standard	Decisional bilinear Diffie–Hellman	√
[23] - \prod_{New}	Standard	Decisional bilinear Diffie–Hellman	√
Ours	Standard	CBDH	√

Table 2. Performance comparison

Scheme	Master public key length	Master secret key length	Secret key length	Ciphertext length
[14]	$ \mathbb{G}_T + (n+3) \mathbb{G}_{p_1} + \mathbb{G}_{p_3} $	$(n+3)(\mathbb{G}_{p_1} \cdot \mathbb{G}_{p_3})$	$(n+2)(\mathbb{G}_{p_1} \cdot \mathbb{G}_{p_3})$	$ \mathbb{G}_T + (n+2) \mathbb{G}_{p_1} $
[26]	$m \mathbb{G}_T + (m+3) \mathbb{G}_{p_1} + \mathbb{G}_{p_3} $	$3m(\mathbb{G}_{p_1} \cdot \mathbb{G}_{p_3})$	$2m(\mathbb{G}_{p_1} \cdot \mathbb{G}_{p_3})$	$ \mathbb{G}_T + 2m \mathbb{G}_{p_1} $
[23] - \prod	$2 \mathbb{G}_T + 2 \mathbb{G} $	$4 \mathbb{Z}_p $	$4 \mathbb{G} $	$2 \mathbb{G} + 2 \mathbb{G}_T + \mathbb{Z}_q $
[23] - \prod_{New}	$2 \mathbb{G}_T + 2 \mathbb{G} $	$4 \mathbb{Z}_p $	$4 \mathbb{G} $	$3 \mathbb{G} + \mathbb{G}_T + 2 \mathbb{Z}_q $
Ours	$(n+3) \mathbb{G} $	$ \mathbb{Z}_p $	$2n \mathbb{G} $	$3 \mathbb{G} + \eta + \mu$

than the schemes [14, 26]. The lengths of the master secret key and ciphertext are shorter than \prod and \prod_{New} in [23]. In addition, the key leakage ratios of \prod and \prod_{New} in [23] are 1/2 and 3/4 respectively, however our leakage ratio is almost 1.

The schemes [14, 23, 26] and our scheme are implemented under Windows 10 system (Intel(R) Core(TM) i7-6500U CPU 2.50 GHz 8.00 GB RAM) using C++ language. In this process, the configuration file a.param of PBC [28] is adopted, and the length of message is 1024 bits. The operating results are listed below.

From the Table 3 and Fig. 1, the run time of our scheme is faster than [23] except for encryption and decryption time. Furthermore, the total operating time of our scheme is less than the total operating time of schemes [14, 23, 26]. Therefore, considering the performance and leakage resistance ability of the schemes, our scheme has more advantages, and has certain application value.

Table 3. Run time (ms)

Schemes	Setup time	KeyGen time	Enc time	Dec time	UpdateSK time	Total time
[14]	254	77	25	12	75	370
[26]	558	245	76	13	236	1470
[23] - \prod	67	26	26	4	22	147
[23] - \prod_{New}	67	26	30	6	22	153
Ours	43	10	31	13	10	110

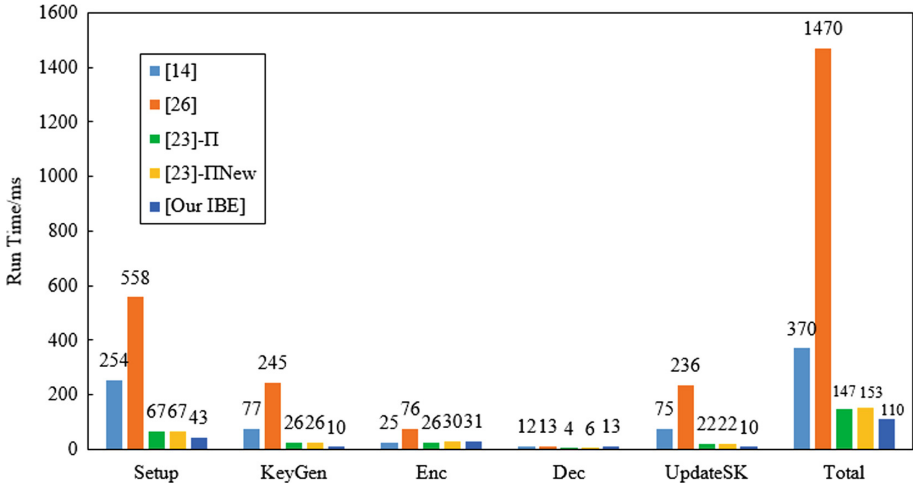


Fig. 1. Run time comparison

7 Conclusions

This paper gives a formal definition and the security model for IBE scheme, which is resilient to the continual leakage. In addition, we construct a concrete IBE scheme resilient to the continual leakage. We prove that this scheme is secure against the adaptive chosen-ciphertext attack in the standard model. The security of the IBE scheme is reduced to the hardness of the computational bilinear Diffie-Hellman problem. Performance analysis is also given. The design of cryptography that can resist leakage is a new research direction. To put forward certain IBE schemes with stronger leakage-resilient property (e.g., random number leakage, after-the-fact leakage, etc.) is our further research work. To construct secure IBE schemes resilient to leakage under different hard problems, such as the lattice hard problem etc. is an open problem.

Acknowledgments. We are thankful to anonymous referees for their helpful comments. This paper is supported by the National Natural Science Foundation of China under Grant No. 61902140, No. 60573026, the Anhui Provincial Natural Science Foundation under Grant No. 1908085QF288, No. 1708085QF154, the Nature Science Foundation of Anhui Higher Education Institutions under Grant No. KJ2018A0398, No. KJ2019A0605, No. KJ2018A0396, No. KJ2019B06.

References

1. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5




2. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
3. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_16
4. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_7
5. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_27
6. Jin, L., Li, J., Chen, X., et al.: Identity-based encryption with outsourced revocation in cloud computing. *IEEE Trans. Comput.* **64**(2), 425–437 (2015)
7. Wu, L., Zhang, Y., Choo, K.K.R., et al.: Efficient identity-based encryption scheme with equality test in smart city. *IEEE Trans. Sustain. Comput.* **3**(1), 44–55 (2017)
8. Lai, J., Mu, Y., Guo, F.: Efficient identity-based online/offline encryption and signcryption with short ciphertext. *Int. J. Inf. Secur.* **16**(3), 1–13 (2017)
9. Zhang, L., Mu, Y., Wu, Q.: Compact anonymous hierarchical identity-based encryption with constant size private keys. *Comput. J.* **59**(4), 452–461 (2016)
10. Micali, S., Reyzin, L.: Physically observable cryptography. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24638-1_16
11. Halderman, J.A., Schoen, S.D., Heninger, N., et al.: Lest we remember: cold-boot attacks on encryption keys. *Commun. ACM* **52**(5), 91–98 (2009)
12. Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_2
13. Dodis, Y., Goldwasser, S., Tauman Kalai, Y., Peikert, C., Vaikuntanathan, V.: Public-key encryption schemes with auxiliary inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 361–381. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_22
14. Lewko, A., Rouselakis, Y., Waters, B.: Achieving leakage resilience through dual system encryption. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 70–88. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_6
15. Halevi, S., Lin, H.: After-the-fact leakage in public-key encryption. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 107–124. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_8
16. Galindo, D., Herranz, J., Villar, J.: Identity-based encryption with master key-dependent message security and leakage-resilience. In: Foresti, S., Yung, M., Martinelli, F. (eds.) ESORICS 2012. LNCS, vol. 7459, pp. 627–642. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33167-1_36
17. Sun, S., Gu, D., Liu, S.: Efficient chosen ciphertext secure identity-based encryption against key leakage attacks. *Secur. Commun. Netw.* **9**(11), 1417–1434 (2016)
18. Li, J., Teng, M., Zhang, Y., et al.: A leakage-resilient CCA-secure identity-based encryption scheme. *Comput. J.* **59**(7), 1066–1075 (2016)
19. Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-key encryption in the bounded-retrieval model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 113–134. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_6

20. Yuen, T.H., Zhang, Y., Yiu, S.M., Liu, Joseph K.: Identity-based encryption with post-challenge auxiliary inputs for secure cloud applications and sensor networks. In: Kutyłowski, M., Vaidya, J. (eds.) ESORICS 2014. LNCS, vol. 8712, pp. 130–147. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11203-9_8
21. Li, J., Yu, Q., Zhang, Y.: Identity-based broadcast encryption with continuous leakage resilience. *Inf. Sci.* **429**(1), 177–193 (2018)
22. Li, J., Guo, Y., Yu, Q., et al.: Provably secure identity-based encryption resilient to post-challenge continuous auxiliary input leakage. *Secur. Commun. Netw.* **9**(10), 1016–1024 (2016)
23. Zhou, Y., Yang, B., Mu, Y., et al.: Continuous leakage-resilient identity-based encryption without random oracles. *Comput. J.* **61**(4), 586–600 (2018)
24. Zhou, Y., Yang, B., Mu, Y.: Continuous leakage-resilient identity-based encryption with leakage amplification. *Des. Codes Cryptogr.* (2019). <https://doi.org/10.1007/s10623-019-00605-0>
25. Goldreich O., Levin A.: A hard-core predicate for all one-way functions. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing-STOC 1989, Washington, pp. 25–32 (1989)
26. Yuen, T.H., Chow, S.S.M., Zhang, Y., Yiu, S.M.: Identity-based encryption resilient to continual auxiliary leakage. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 117–134. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_9
27. Chen, Y., Chen, L., Zhang, Z.: CCA secure IB-KEM from the computational bilinear Diffie-Hellman assumption in the standard model. In: Kim, H. (ed.) ICISC 2011. LNCS, vol. 7259, pp. 275–301. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31912-9_19
28. Lynn, B.: PBC (Pairing-Based Cryptography) Library (2012). <http://crypto.stanford.edu/pbc/>

Post-quantum Cryptography



CLIBDA: A Deniable Authentication Scheme for Pervasive Computing Environment

Emmanuel Ahene¹ , Yuanfeng Guan², Zhiwei Zhang², and Fagen Li¹  

¹ School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
eahene@gmail.com , fagenli@uestc.edu.cn

² SI-TECH Information Technology Co. Ltd., Beijing, China

Abstract. Pervasive computing environments permits users to get the services they require at anywhere and anytime. Security turns to be a major challenge in pervasive computing environments due to its heterogeneity, dynamicity, mobility and openness. In this paper, we propose a new heterogeneous deniable authentication scheme called CLIBDA for pervasive computing environments utilizing bilinear pairings. The proposed CLIBDA scheme permits a sender in certificateless cryptography (CLC) setting to transmit a message securely to a receiver in an identity based cryptography (IBC) setting. Detailed security analysis shows that the CLIBDA scheme is secure in the random oracle model (ROM) under the bilinear Diffie–Hellman assumption. Additionally, CLIBDA supports batch verification which is necessary for the speed up of the verification of authenticators. This characteristic makes the CLIBDA scheme suitable in pervasive computing environments.

Keywords: Pervasive computing · Deniable authentication · Authentication · Heterogeneity · Security

1 Introduction

Pervasive computing [1, 2] is a growing trend of advanced computing that gives computational capabilities to different objects or mobile devices and resultantly empowers them to effectively assure communication and perform tasks. Pervasive computing promises to simplify daily life via the integration of mobile devices and digital infrastructures into our real world. This computing technology enables users to interact with systems using laptop computers, mobile phones, tablets and also terminals in everyday objects like a pair of glasses or refrigerator. That means in a pervasive computing environment, users can get services they require at anytime and anywhere. The underlying technologies which mainly supports

This work is supported by the National Natural Science Foundation of China (grant no. 61872058).

pervasive computing comprise Internet, microprocessors, advanced middleware, wireless communication, operating systems, sensors, cloud computing and so on. While pervasive computing renders convenient access to pertinent information and applications, it poses several research challenges. Security turns up to be one of the vital issues due to its heterogeneity, dynamicity, mobility and openness. Specifically, authentication is a pertinent security requirement for the pervasive computing environments holding to the need for service providers to authenticate users and additionally be certain that they are accessing their legitimate services in a valid manner [3]. Some fitting authentication schemes relative to pervasive computing environments are proposed in [3–8] and generally, we see from these schemes that in pervasive environments, often users may send an evidence to service providers and the service providers may check its validity. If the evidence turns out valid, the service provider enables the user's access right to the service. Otherwise, the user's access request is rejected. Via authentication we obtain non-repudiation which prevents the denial of a previous action. However, not all pervasive computing-enabled applications require non-repudiation. Some applications like electronic voting [9], online haggling, secure online negotiation [10] and e-mail [11] do not desire non-repudiation, instead they desire to have deniability, and this security property is obtained from deniable authentication (DA) schemes. A DA scheme is characterized by these features. (1) It gives an intended receiver to the ability to identify the source of a given message. (2) Even under coercion, an intended receiver can not successfully prove the source of given message to a third party. Nonetheless, these features are imperative for online haggling, secure online negotiation and additionally electronic voting systems. For this cause, we propose a new heterogeneous DA scheme called CLIBDA that is suitable for applications that require deniability in pervasive computing environments.

1.1 Related Work

In descending order of their emergence; certificateless cryptosystem (CLC), public identity-based cryptosystem (IBC) and key infrastructure (PKI) are three known public key cryptosystems. In a PKI, there exist a certificate authority (CA) responsible to issue public key certificate. The reason for the certificate is to assure the binding of the public key and the user's identity via the signature of CA. Via the verification of the public key certificate, we are able to ascertain the validity of a public key. For a valid certificate, we say the public key is also essentially valid. Nonetheless, the major difficulty in PKI is about how to efficiently manage public key certificates, including their storage, distribution, revocation, and cost (computational) of certificate verification. To eradicate the reliance on these public key certificates and to additionally simplify public key management, the concept of IBC emerged [12]. In IBC, a user's public key is directly computable from its identity information, for instance, e-mail address, IP address and telephone number. The validity of a public key is verifiable without public key certificate. Nonetheless, the corresponding private key generated is acquired from some trusted third party identified as private key generator (PKG).

However, this makes IBC weak against key escrow attack. Consequently, CLC emerged as a remedy to the key escrow shortfall in IBC [13]. As a remedy, CLC requires the trusted third party named as key generator center (KGC) to only generate a partial private key utilizing the master private key. In essence a user may obtain full private key via the combination a secret key (chosen by the user) with the partial private key. Here, the corresponding public key is generated via combining the user's secret key with system parameters. The CLC has neither the shortfall in PKI nor the key escrow shortfall (since the KGC remains unaware of the user's generated secret key).

Following after these three public key cryptosystems, researchers have designed many useful DA schemes in PKI setting [14–19], deniable authentication schemes in the IBC setting [20–22] and DA schemes in CLC setting [23, 24]. In [14], Wang and Song came up with a non-interactive DA scheme based on designated verifier proofs. Additionally, in [15], Raimondo and Gennaro designed two approaches for DA. Their schemes do not need the use of a CCA-secure encryption; essentially, they demonstrated a distinct generic approach to the problem of DA. Tian et al. [16] developed a new paradigm for the construct of non-interactive DA schemes with a well structured security proof mechanism. One similarity among schemes in [16–19] is that they are all PKI-based DA schemes and hence they possess the inherent certificate management burden. To improve on the state-of-the art, Lu et al. [20] developed an ID-based DA scheme which was proven secure under the RSA assumption. Li et al. [21] proposed an efficient DA scheme and proved its security in the ROM. In [23], Jin et al. designed a pairing based CL deniable authentication scheme under the CDH and BDH assumptions (hereafter called JXLZ). Later, Jin et al. [24] came up with a non-pairing based CL deniable authentication protocol formally proved it in the ROM (hereafter called JXZL).

Common among the aforementioned DA schemes is homogeneity (communication parties run in the same environment). For instance, in these schemes [14–19], communication parties run in PKI environment; in these schemes [20–22], communication parties run in IBC environment; while communication parties in these schemes [23, 24] run in CLC environment. This characteristic of homogeneity makes them unsuitable for pervasive systems. In [25], Li et al. designed two heterogeneous DA (HDA) schemes for pervasive computing environments. The first HDA scheme permits a PKI-based sender to send a message to an IBC-based receiver. The second HDA permits an IBC-based sender to send a message to a PKI-based receiver. Moreover, in [26], Jin et al. designed a heterogeneous scheme which permits a CLC-based sender to send a message to a PKI-based receiver (hereafter called JCYZ).

1.2 Contribution

Because almost all existing DA schemes are homogeneous, they remain unsuitable for pervasive computing environments. Although heterogeneous DA schemes have been proposed, none of them provides a solution that enables CLC-based users to interact with IBC-based users in a pervasive computing environment.

The contribution of our paper is to address this design problem or gap in HDA and hence pervasive computing environments. Specifically, we propose an HDA scheme that enables a sender in CLC environment to interact or send a message to a receiver in IBC environment. We name our scheme as CLIBDA and formally define its security model. Moreover, we prove its security in the ROM under the BDH assumptions. Additionally, CLIBDA provides batch verification, which is well needed in several applications in pervasive computing environment for fast verification of authenticators. CLIBDA is suitable for applications such as online haggling, secure online negotiation and additionally electronic voting systems.

1.3 Organization

The remainder of this paper follows thusly. Preliminaries are presented in Sect. 2. While CLIBDA scheme is given in Sect. 3. We present an analyses of its security in Sect. 4 and present on CLIBDA scheme's performance and application in Sect. 5. Section 6 provides conclusion.

2 Preliminaries

2.1 Bilinear Pairings

Let G_1 be a symbolic name of a cyclic group (additive) that has prime order q . Again, G_2 means a cyclic group (multiplicative) that is identified with same order q . Let P be named as a generator of G_1 . Consequently, a map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ is described as bilinear pairing reliant on these properties:

1. Bilinearity: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ where $P, Q \in G_1$ and $a, b \in \mathbb{Z}_q^*$.
2. Non-degeneracy: $\hat{e}(P, Q) \neq 1$ given that $P, Q \in G_1$ and the identity element in G_2 is 1.
3. Computability: For all $P, Q \in G_1$, a plausible algorithm exist to aptly compute $\hat{e}(P, Q)$.

The permissible map \hat{e} is gotten typically from either Tate or modified Weil pairings [27, 28]. Our scheme's security relies on the following defined intractable problems.

Given G_1, G_2, P , and $\hat{e}: G_1 \times G_1 \rightarrow G_2$, The bilinear Diffie-Hellman (BDH) problem in (G_1, G_2, \hat{e}) is to produce $Y = \hat{e}(P, P)^{abc}$ given (P, aP, bP, cP) . Here, $a, b, c \in \mathbb{Z}_q^*$.

Definition 1. The (ϵ, t) -BDH assumption holds when no t -polynomial time adversary \mathcal{C} possess an advantage of at least ϵ in solving the BDH problem.

3 A New Heterogeneous Deniable Authentication Scheme

Here, we formally define our CLIBDA scheme and subsequently define its security notions. We have defined all related symbols in Table 1.

Table 1. Notations.

Symbol	Description
s	Master secret key
λ	Security parameter
\hat{e}	A bilinear map
G_1	Additive cyclic group
G_2	Multiplicative cyclic group
q	Large prime number
P	Generator of G_1
H_i	Hash function, where $i = 1, 2, 3, 4, 5$
P_{pub}	Master public key
ID	An CLC entity's identity but also public key for an IBC entity
\perp	Error symbol
S_{ID}	Private key
D_{ID}	Partial private key
x_{ID}	An entity's generated secret value
Pk_{ID}, K_{ID}	Partial public keys of entities in CLC environment
P_{ID}	An entity's public key where $P_{ID} = (Pk_{ID}, K_{ID})$
m	Message
σ	Deniable authenticator

3.1 Syntax

A generic CLIBDA scheme comprise six algorithms:

- **Setup:** The KGC utilizes a security parameter λ in this algorithm as an input and then produce as an output a master secret key s , system parameters $param$ including master public key P_{pub} . We suppose that $param$ are made public and are implicit inputs in subsequent algorithms.
- **Extract partial private key:** The KGC utilizes s and user identity ID_u in this algorithm as input and then produce as an output to user, a partial private key D_u . Here, u signifies either a sender or receiver.
- **User key generation:** The user utilizes ID_u in this algorithm as input and then produces as output a secret value x_u and additionally public key P_u . P_u is published devoid of certification.
- **Private key generation:** The user utilizes D_u and the user-created secret value x_u in this algorithm as input and then produces an output of private key S_u .
- **IB-KE:** The user in the identity based setting sends its ID_r to the KGC. The KGC returns corresponding private key S_r to the user.
- **Authenticate:** Using as input a message m , ID_s , S_s , P_s , and ID_r identified respectively as sender's identity, private key, public key and receiver's public

key, the algorithm produces as an output a deniable authenticator σ . The sender executes this probabilistic algorithm.

- **Verify**: Using as input σ, ID_s, P_s, ID_r , and S_r identified respectively as the deniable authenticator, sender’s identity, public key, a receiver’s identity and private key, the algorithm produces an output of \top for a valid σ or otherwise \perp indicating the invalidity of σ . This deterministic algorithm is executed by the receiver.

A secure CLIBDA scheme must comply by the algorithms above and their respective definitions. Additionally, all algorithms must satisfy the given CLIBDA consistency constraint thusly:

If $\sigma = \text{Authenticate}(m, S_s, P_s, ID_s, ID_r)$
 then $m = \text{Verify}(\sigma, S_r, P_s, ID_r)$
 Here, $params$ is excluded for simplicity.

3.2 Security Notions

First, the CLIBDA scheme must accomplish deniable authentication (DA). Specifically, deniable authentication against adaptive chosen message attacks (DA-CMA) [21]. Here, we apply slight modification to the security model in Li et al.’s [21] protocol to make it fitting for the CLIBDA scheme. We utilize the adversarial model (Type I and Type II) as in [13] since the sender in CLIBDA scheme is in the CLC setting. A Type I adversary is ineligible to access the master secret key, however, it is eligible to substitute a user’s public key with a preferred valid public key. A Type II adversary is ineligible to substitute a user’s public key, however, it is eligible to access the master secret key. Hereafter, we use the sequence of games to describe the security notions for the CLIBDA scheme. Each game is played by a challenger \mathcal{C} and a certain adversary say \mathcal{F}_I or \mathcal{F}_{II} .

Game I: In this game \mathcal{F}_I interacts with \mathcal{C} thusly:

- **Setup**: \mathcal{C} initiates **Setup** algorithm taking as input λ and produces $params$ to \mathcal{F}_I .
- **Attack**: \mathcal{F}_I runs a number queries that are polynomially bounded adaptively.
- **Partial private key queries**: \mathcal{F}_I queries with identity ID . \mathcal{C} calls **Extract partial private key** algorithm and resultantly sends D_{ID} to \mathcal{F}_I as partial private key.
- **Private key queries**: \mathcal{F}_I queries with identity ID . \mathcal{C} calls **Private key generation** algorithm and resultantly sends S_{ID} to \mathcal{F}_I as the full private key. Observe that, \mathcal{C} may initially call **Extract partial private key** algorithm if necessary.
- **Public key queries**: \mathcal{F}_I queries with identity ID . \mathcal{C} calls **User key generation** algorithm and resultantly sends P_{ID} as public key to \mathcal{F}_I .
- **Public key replacement queries**: \mathcal{F}_I may resolve to substitute P_{ID} with a preferred valid public key.
- **Key extraction queries**: \mathcal{F}_I queries with identity ID . \mathcal{C} calls **IB-KE** algorithm and resultantly sends S_{ID} to \mathcal{F}_I as the full private key.

- *Deniable authentication queries:* \mathcal{A}_I queries with the triple (m, ID_s, ID_r) . \mathcal{C} first initialize **Private key generation** and **User key generation** algorithms to respectively acquire the private key S_s and additionally the sender’s public key P_s . \mathcal{C} then calls **Authenticate** $(m, S_s, P_s, ID_s, ID_r)$ and sends σ to \mathcal{F}_I . Here, note that we first expect \mathcal{F}_I to reveal any secret value, if the associated public key has been replaced.
- *Verify queries:* \mathcal{F}_I queries with the triple (σ, ID_s, ID_r) . \mathcal{C} initially calls **IB-KE** algorithm for the private key S_r of the receiver. \mathcal{C} then executes **Verify** $(\sigma, S_r, ID_s, P_s, ID_r)$ and sends the resulting value to \mathcal{F}_I . The value is either \top for a valid σ or otherwise \perp indicating the invalidity of σ .

Forgery: \mathcal{F}_I generates $(m^*, ID_r^*, ID_s^*, \sigma^*)$ identified as message, receiver’s identity, sender’s identity and deniable authenticator respectively. At last \mathcal{F}_I successfully wins given that these conditions apply:

1. **Verify** $(\sigma^*, S_r^*, ID_s^*, P_s^*, ID_r^*) = \top$.
2. \mathcal{F}_I has not run private key query with ID_s^* or key extraction query with ID_r^* .
3. \mathcal{F}_I has not run public key replacement and partial private key query with ID_s^* .
4. \mathcal{F}_I has not run deniable authentication query with (m, ID_r^*, ID_s^*) .
5. \mathcal{F}_I has not run verify query with (σ, ID_r^*, ID_s^*) .

The advantage of \mathcal{F}_I relies on its probability to succeed.

Definition 2. A CLIBDA scheme is $(\epsilon_{da}, t, q_k, q_{pk}, q_{ppk}, q_d, q_v)$ -Type-I-DA-CMA secure when a polynomially bounded adversary \mathcal{F}_I possesses a negligible advantage to succeed with an advantage of ϵ_{da} after at most q_k key extraction queries, q_{ppk} partial private key queries, q_{pk} public key queries, q_d deniable authentication queries and additionally q_v verify queries.

Game II: In this game \mathcal{F}_{II} interacts with \mathcal{C} thusly:

- **Setup:** \mathcal{C} initiates **Setup** algorithm taking as input λ and produces $params$ and s to \mathcal{F}_{II} .
- **Attack:** \mathcal{F}_{II} runs a number queries that are polynomially bounded adaptively as in **Game I** exclusive of partial private key queries. It follows after the reason that \mathcal{F}_{II} has s and can develop sender’s partial private key by itself.

Forgery: \mathcal{F}_{II} generates a triple m^*, ID_r^*, σ^* identified as message, receiver’s identity and deniable authenticator respectively. At last \mathcal{F}_{II} successfully wins given that these conditions apply:

1. **Verify** $(\sigma^*, S_r^*, ID_s^*, P_s^*, ID_r^*) = \top$.
2. \mathcal{F}_{II} has not run private key query with ID_s^* .
3. \mathcal{F}_{II} has not run deniable authentication query with (m, ID_r^*, ID_s^*) .
4. \mathcal{F}_{II} has not run verify query with (σ, ID_r^*, ID_s^*) .

The advantage of \mathcal{F}_{II} relies on its probability to succeed.

Definition 3. A CLIBDA scheme is $(\epsilon_{da}, t, q_{sk}, q_{pk}, q_d, q_v)$ -Type-II-DA-CMA secure when a polynomially bounded adversary \mathcal{F}_{II} possesses a negligible advantage to succeed with an advantage of ϵ_{da} after at most q_{sk} private key queries, q_{pk} public key queries, a q_d authentication queries and additionally q_v verify queries.

Definition 4. A CLIBDA scheme is DA-CMA secure when a polynomially bounded adversary possesses a negligible advantage in the respective Type I and Type II games.

Observing Game I and Game II exposes how the adversary is kept unaware of S_r^* (the receiver's private key). This corresponds well to the deniability property for which the sender can deny its actions done. Due to the receiver's ability to also generate a deniable authenticator which is valid. This is the major distinction between deniable authentication and the undeniable authentication (unforgeability) in digital signature schemes.

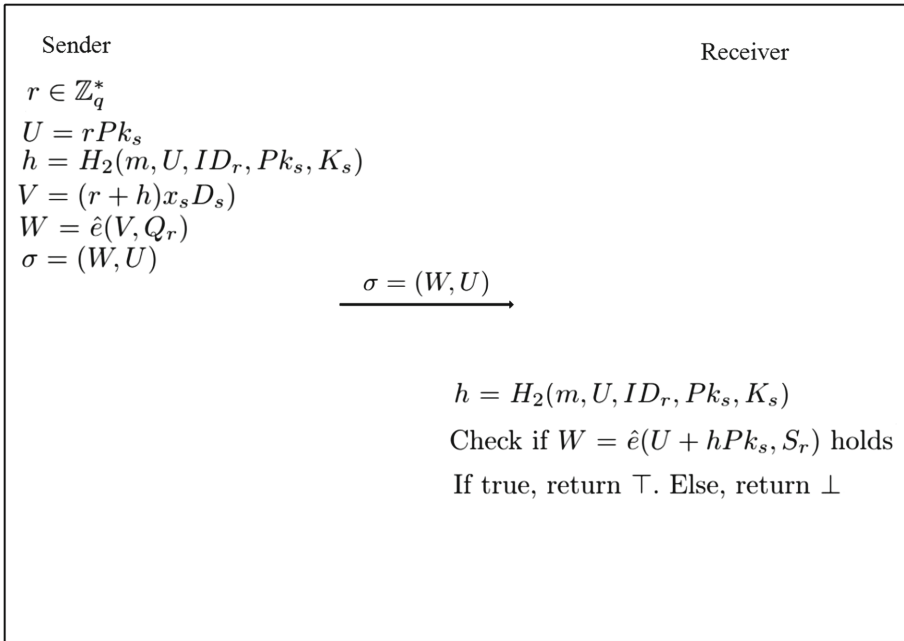


Fig. 1. A CLIBDA scheme.

3.3 CLIBDA Scheme

As shown in Fig. 1, we propose a heterogeneous deniable authentication scheme on pairings with these algorithms:

- **Setup:** Given λ as a security parameter, this algorithm generates (G_1, G_2, \hat{e}) where $\hat{e} : G_1 \times G_1 \rightarrow G_2$ is a bilinear pairing and G_1 and G_2 are two groups of the same prime order q such that $q \geq 2^\lambda$. To determine the length of q we set $\lambda \geq 160$. Discrete logarithm problem (DLP) is known to be harder in G_1 and G_2 for such values of λ . The algorithm continues to work as follows:
 1. Randomly selects $s \in \mathbb{Z}_q^*$ as master secret key and derives the master public key $P_{pub} = sP$ where $P \in G_1$ is a chosen arbitrary generator. s is kept as a secret by the KGC.
 2. Chooses hash functions; $H_1 : \{0, 1\}^* \rightarrow G_1$, and $H_2 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$.
 3. Publishes the public parameters $params = (q, \hat{e}, G_1, G_2, P, P_{pub}, H_1, H_2)$.
- **Extract partial private key:** A user sends his identity ID_u to the KGC in request for a partial private key. The KGC runs this algorithm as follows:
 1. Computes $Q_u = H_1(ID_u)$.
 2. Computes the partial private key $D_u = sQ_u$ where s is master secret key. The KGC sends D_u to the user.
- **User key generation:** The user randomly selects $x_u \in \mathbb{Z}_q^*$ as a secret value and computes public keys $K_u = x_u P$ and $Pk_u = x_u Q_u$. Hence $P_u = (Pk_u, K_u)$.
- **Private key generation:** The user executes this algorithm. The algorithm takes the D_u obtained from the KGC and the randomly chosen secret value x_u as input and then generates a private key $S_u = (x_u, D_u)$.
- **IB-KE:** Given an identity ID_u , the KGC computes the corresponding private key $S_u = sQ_u$ and sends it to its owner in a secure way.
- **Authenticate:** Given the message m , Alice's identity ID_s , private key S_s , public key P_s , and additionally the public key ID_r of Bob, this algorithm proceeds thusly:
 1. Choose a random $r \in \mathbb{Z}_q^*$ and compute $U = rPk_s$.
 2. Compute $h = H_2(m, U, ID_r, Pk_s, K_s)$.
 3. Compute $V = (r + h)x_s D_s$.
 4. Compute $W = \hat{e}(V, Q_r)$.
 5. Compute $\sigma = (W, U)$.
- **Verify:** Upon inputting a message m , a deniable authenticator σ , a sender's identity ID_s and public key $P_s = (Pk_s, K_s)$, a receiver's private key S_r and public key ID_r , the receiver executes the following procedures.
 1. Compute $h = H_2(m, U, ID_r, Pk_s, K_s)$.
 2. Check if $W = \hat{e}(U + hPk_s, S_r)$ holds. If true, return \top . Else, return \perp .

3.4 Consistency

Our CLIBDA scheme's consistency can be ascertained with bilinearity thusly:

$$\begin{aligned}
 W &= \hat{e}(U + hPk_s, S_r) \\
 &= \hat{e}(rPk_s + hPk_s, S_r) \\
 &= \hat{e}((r + h)Pk_s, S_r) \\
 &= \hat{e}((r + h)x_s Q_s, sQ_r) \\
 &= \hat{e}((r + h)x_s sQ_s, Q_r) \\
 &= \hat{e}((r + h)x_s D_s, Q_r) \\
 &= \hat{e}(V, Q_r)
 \end{aligned}$$

The proposed CLIBDA scheme also has support for batch verification. For instance, given n deniable authenticators

$$(P_1, m_1, \sigma_1), \dots, (P_n, m_n, \sigma_n) \quad (1)$$

where $\sigma_i = (W_i, U_i)$ and $P_i = (Pk_i, K_i)$ for all $(i = 1, \dots, n)$. A receiver with S_r as secret key runs verification for these authenticators in a simultaneous manner by computing $h_i = H_2(m_i, U_i, ID_{r_i}, Pk_{s_i}, K_{s_i})$ for all $(i = 1, 2, \dots, n)$ and checking whether

$$\prod_{i=1}^n W_i = \hat{e}\left(\sum_{i=1}^n U_i + \sum_{i=1}^n h_i Pk_i, S_r\right) \quad (2)$$

holds. If true, return \top . Else, return \perp .

4 Security Analysis

We affirm that the CLIBDA is deniable via Theorem 1 and that it is DA-CMA secure via Theorem 2.

4.1 Deniability

Theorem 1. *The proposed CLIBDA scheme is deniable.*

Proof. On the receipt of a deniable authenticator $\sigma = (W, U)$, the receiver may discover the source of m (the given message) with S_r (its private key). The receiver may capably simulate the transcripts of a m thusly:

1. Choose randomly an $\bar{r} \in \mathbb{Z}_q^*$ and compute $\bar{U} = \bar{r}Pk_s$
2. Compute $\bar{h} = H_2(m, \bar{U}, ID_r, Pk_s, K_s)$.
3. Compute $\bar{W} = \hat{e}(\bar{U} + \bar{h}Pk_s, S_r)$

Now $\bar{\sigma} = (\bar{W}, \bar{U})$ can be generated with ease by the receiver. This $\bar{\sigma}$ remains indistinguishable from the $\sigma = (W, U)$ of the sender via **Authenticate** algorithm. Given that $\hat{\sigma} = (\hat{W}, \hat{U})$ is a valid deniable authenticator chosen randomly from the set of all outcomes of valid deniable authenticators intended for the receiver then essentially the probability $\Pr[\bar{\sigma} = \hat{\sigma}] = 1/(q - 1)$ since $\bar{\sigma}$ is deduced from a selected random value $\bar{r} \in \mathbb{Z}_q^*$. Similarly, $\Pr[\sigma = \hat{\sigma}] = 1/(q - 1)$ because it is deduced from $r \in \mathbb{Z}_q^*$. In essence, we obtain the same probability distributions.

Theorem 2. *In the ROM, our scheme is DA-CMA secure against the adversary \mathcal{F}_I or the adversary \mathcal{F}_{II} supposing the BDH assumption is intractable.*

Proof. We prove Theorem 2 via Lemmas 1 and 2.

Lemma 1. Under the ROM, suppose adversary \mathcal{F}_I exists that is equipped to break the Type-I-DA-CMA security of our CLIBDA scheme, running at time t and initiating at most q_{sk} private key queries, q_{pk_r} public key replacement queries, q_{ppk} partial private key queries, q_{pk} public key queries, q_d deniable authentication queries, q_v verify queries and additionally q_{H_i} oracle (for H_i ($i = 1, 2$)) queries with an advantage $\epsilon_{da} \geq 10(q_d + 1)(q_d + q_{H_2})q_{H_1}/(2^\lambda - 1)$, then an algorithm \mathcal{C} exists that is adept to solve the BDH problem at expected time $t' \leq 120686q_{H_2}q_{H_1}2^{\lambda t}/(2^\lambda - 1)$.

Proof. We prove this Lemma 1 utilizing forking lemma [29]. To adopt the forking lemma, we first explain how our scheme fits into the signature scheme given in [29], the simulation step for which the deniable authenticator can be well simulated without the sender's private key (additionally without the master secret key), and how the BDH problem can be solved based on the forgery.

First, we observe that the tuple (σ_1, h, σ_2) is produced during the deniable authentication of message m , which holds to correspond to the required three-phase honest-verifier zero-knowledge identification protocol. Here, $\sigma_1 = U$ is the commitment of the prover, $h = H_2(m, U, ID_r, Pk_s, K_s)$ is the hash value substituting the verifier's challenge (h relies on m and σ_1) and $\sigma_2 = W$ represents the prover's response. Furthermore, we proceed to show the simulation steps that renders a faithful simulation of a forger \mathcal{F}_I and how to get the answer to the BDH problem via interacting with \mathcal{F}_I . With a given instance (P, aP, bP, cP) , \mathcal{C} aims to compute $\hat{e}(P, P)^{abc}$. \mathcal{C} interacts with \mathcal{F}_I responding appropriately to the queries of \mathcal{F}_I via the random oracles H_1 and H_2 . To monitor H_1 and H_2 queries, \mathcal{C} keeps up two records L_1 and L_2 for storage of hash oracle answers respectively. Additionally, \mathcal{C} keeps record L_3 of the private keys and public keys it may have given out. By assumption H_1 queries are considered as distinct and that \mathcal{F}_I may ask for $H_1(ID)$ prior to the use of ID in other queries. We present the processes thusly:

- **Setup:** \mathcal{C} initiates **Setup** algorithm taking as input λ and produces $params$ to \mathcal{F}_I where $P_{pub} = cP$.
- **Attack:** \mathcal{F}_I runs a number queries that are polynomially bounded adaptively.
- H_1 queries: \mathcal{C} first chooses randomly $a, b \in \{1, 2, \dots, q_{H_1}\}$. \mathcal{F}_I runs a number of H_1 queries that are polynomially bounded on identities it prefers. At the ℓ -th H_1 query, \mathcal{C} replies $H_1(ID_\ell) = bP$ and at the γ -th H_1 query, \mathcal{C} replies $H_1(ID_\ell) = aP$. For all other $H_1(ID_i)$ queries where $i \neq \ell$, \mathcal{C} picks at random $d_i \in \mathbb{Z}_q^*$, adds (ID_i, d_i) to L_1 and submit $H_1(ID_i) = d_iP$ to \mathcal{F}_I as response.
- H_2 queries: The \mathcal{F}_I queries with $H_2(m, U, ID_r, Pk_s, K_s)$, \mathcal{C} replies by first checking whether or not H_2 was pre-defined for $H_2(m, U, ID_r, Pk_s, K_s)$. If it was so, then the pre-defined value is returned. Else, \mathcal{C} replies with $h \in \mathbb{Z}_q^*$ to \mathcal{F}_I and inserts the tuple $(m, U, ID_r, Pk_s, K_s, h)$ into the record L_2 .
- *Partial private key queries:* \mathcal{F}_I queries with identity ID_i . For $ID_i = ID_\ell$ or $ID_i = ID_\gamma$, \mathcal{C} terminates. For $ID_i \neq ID_\ell, ID_\gamma$, \mathcal{C} checks up record L_1 and returns $D_i = d_i cP$ and respectively updates record L_3 with $(ID_i, \perp, D_i, \perp)$.

- *Public key queries:* \mathcal{F}_I queries with identity ID_i , to reply, \mathcal{C} looks for tuple $(ID_i, D_i, Pk_i, K_i, x_i)$ in L_3 . If such a tuple is in L_3 , then \mathcal{C} produces to \mathcal{F}_I both PK_i and K_i . Else \mathcal{C} picks randomly $x_i \in \mathbb{Z}_q^*$, outputs $PK_i = x_i d_i P$ and $K_i = x_i P$, updates L_3 with $(ID_i, \perp, Pk_i, K_i, x_i)$ and sends to \mathcal{F}_I both PK_i and K_i . Note, D_i is acquired only after *Partial private key queries* with ID_i .
- *Private key queries:* \mathcal{F}_I queries with identity ID_i , If no valid replacement of ID_i 's public key has been done and $ID_i \neq ID_\ell, ID_\gamma$ then \mathcal{C} checks L_3 and replies \mathcal{F}_I with $S_i = (x_i, D_i)$. If a valid replacement of ID_i 's public key has been done, then \mathcal{F}_I will in this vain not be given S_i . Observe that, for any identity (ID_ℓ included), \mathcal{F}_I can initiate a replacement of public key. Thus for any identity (ID_ℓ included), \mathcal{F}_I remains aware of their secret value correspondingly. If $ID_i = ID_\ell$ or $ID_i = ID_\gamma$ then \mathcal{C} terminates.
- *Public key replacement queries:* \mathcal{F}_I queries with ID_i for valid replacement of public keys Pk_i and K_i with Pk'_i and K'_i . \mathcal{C} updates L_3 with $(ID_i, \perp, D_i, Pk'_i, K'_i)$. The new value is useful for \mathcal{C} 's subsequent computations.
- *Key extraction queries:* \mathcal{F}_I queries with identity ID_i . For $ID_i = ID_\ell, \mathcal{C}$ terminates. For $ID_i \neq ID_\ell, \mathcal{C}$ checks up record L_1 and returns $S_i = d_i c P$.
- *Deniable authentication queries:* The sender is identified by ID_s and ID_r , symbolizes a receiver's identity. Let m symbolize the message for deniable authentication purposes. When \mathcal{F}_I queries with (m, ID_s, ID_r) , \mathcal{C} replies thusly:
 - If $ID_s \neq ID_\ell, ID_\gamma$ then \mathcal{C} runs at first the private key query for S_s and the public key oracle to acquire Pk_s and K_s . \mathcal{C} replies \mathcal{F}_I at the run of **Authenticate** algorithm.
 - If $ID_s = ID_\ell$ or $ID_s = ID_\gamma$, \mathcal{C} picks randomly $r, h \in \mathbb{Z}_q^*$, puts $U = rP - hPk_s, V = rP_{pub}$ and defines $H_2(m, U, ID_r, Pk_s, K_s) = h$, computes $W = \hat{e}(V, Q_r)$. \mathcal{C} fails when $H_2(m, U, ID_r, Pk_s, K_s)$ is predefined. The \mathcal{C} 's probability of failure is $(q_{da} + q_{H_2})/2^\lambda$. Finally, $\sigma = (U, W)$ is resultantly returned to \mathcal{F}_I by \mathcal{F}_I .
- *Verify:* The \mathcal{F}_I queries with message $m, \sigma = (U, W), ID_s$, and ID_r . \mathcal{C} replies thusly:
 - If $ID_r \neq ID_\ell, ID_\gamma$ then \mathcal{C} runs at first the key extraction query for S_r and \mathcal{C} replies \mathcal{F}_I at the run of **Verify** algorithm.
 - If $ID_r = ID_\ell$ or $ID_s = ID_\gamma$ \mathcal{C} fails and terminates. Obviously the probability of fail at verification queries is at most $q_v = 2^\lambda$.

Forgery: \mathcal{F}_I generates $(m^*, ID_r, ID_s, \sigma^*)$ identified as message, receiver's identity, sender's identity and deniable authenticator respectively. The forking lemma [29] is fitting for identity-less chosen message attack, so we combine the message m^* and the identities $ID_c = \{ID_r, ID_s\}$ as a *generalized* forged message (ID_c, m^*) . By this approach, we disguise the identity-based facet of the DA-CMA attacks and then simulate the settings of an identity-less adaptive-CMA

existential forgery in which case the forking lemma is proven. As in forking lemma, assuming \mathcal{F}_I is an efficient forger then we can legitimately construct a Las Vegas machine \mathcal{F}'_I that generates two deniable authenticators $((ID_c, m^*), h^*, W^*)$ and $((ID_c, m^*), \bar{h}^*, \bar{W}^*)$ with $h \neq \bar{h}^*$ and same commitment U^* . To get the solution to solve the given BDH problem utilizing the machine \mathcal{F}'_I obtained from \mathcal{F}_I , we create a machine \mathcal{C}' thusly.

1. \mathcal{C}' runs \mathcal{F}'_I to obtain two distinct deniable authenticators $((ID_c, m^*), h^*, W^*)$ and $((ID_c, m^*), \bar{h}^*, \bar{W}^*)$.
2. \mathcal{C}' provides the answer to the BDH problem by computing $\hat{e}(P, P)^{abc} = (W^*/\bar{W}^*)^{1/(h^*-\bar{h}^*)}$.

From forking lemma and the lemma in [27], if \mathcal{F}_I is successful in time t with probability $\epsilon_{da} \geq 10(q_d + 1)(q_d + q_{H_2})q_{H_1}/(2^\lambda - 1)$, then an algorithm \mathcal{C} exists that is adept to solve the BDH problem at expected time $t' \leq 120686q_{H_2}q_{H_1}2^\lambda t/(2^\lambda - 1)$.

Lemma 2. Under the ROM, suppose adversary \mathcal{F}_{II} exists that is equipped to break the Type-II-DA-CMA security of our CLIBDA scheme, running at time t and initiating at most q_{sk} private key queries, q_{pk} public key queries, q_d deniable authentication queries, q_v verify queries and additionally q_{H_i} oracle (for H_i ($i = 1, 2$)) queries with an advantage $\epsilon_{da} \geq 10(q_d + 1)(q_d + q_{H_2})q_{H_1}/(2^\lambda - 1)$, then an algorithm \mathcal{C} exists that is adept to solve the BDH problem at expected time $t' \leq 120686q_{H_2}q_{H_1}2^\lambda t/(2^\lambda - 1)$.

Proof. Similar to **Game 1**, we adopt the forking lemma [29] to determine how the BDH problem can be solved based on the forgery in **Game 2**. We follow after the same assumptions in Lemma 1 and exhibit how \mathcal{C} interacts with \mathcal{F}_{II} thusly:

- **Setup:** \mathcal{C} initiates the **Setup** algorithm utilizing λ and returns to \mathcal{F}_{II} system parameters $params$ where $P_{pub} = sP$. s is randomly chosen by \mathcal{C} .
- **Attack:** \mathcal{F}_{II} issues sequences of polynomially bounded queries similar to those queries in **Game 1** with the exception of partial private key queries.
- **Forgery:** Same as in **Game 1**.

From forking lemma and the lemma in [27], if \mathcal{F}_{II} is successful in time t with probability $\epsilon_{da} \geq 10(q_d + 1)(q_d + q_{H_2})q_{H_1}/(2^\lambda - 1)$, then an algorithm \mathcal{C} exists that is adept to solve the BDH problem at expected time $t' \leq 120686q_{H_2}q_{H_1}2^\lambda t/(2^\lambda - 1)$.

5 Performance

Table 2. Performance comparison

Scheme	Computational cost			Communication overhead	Heterogeneous system
	Authenticate	Verify	Batch verify		
JXLZ [23]	1P + 3M	1P + 2M	N/A	$ G_2 + m + \mathbb{Z}_q^* $	N/A
JXZL [24]	7M	6M	N/A	$ m + 2 \mathbb{Z}_q^* $	N/A
JCYZ [26]	1P + 3M	1P + 2M	1P + 2(n-1)M	$ G_2 + m + G_1$	Yes
CLIBDA	1P + 2M	1P + 1M	1P + (n-1)M	$ G_2 + m + G_1$	Yes

We assess our CLIBDA scheme’s computation cost and communication overhead as against JXLZ [23], JXZL [24] and JCYZ [26] as indicated in Table 2. From Table 2, M means point multiplication in G_1 , P means pairing operation whereas N/A means not applicable. All operations excluding those indicated in Table 2 are overlooked because they have negligible effect against the performance metrics. Observe in Table 2 that the computational cost of our CLIBDA scheme is lesser than JXLZ [23] and JCYZ [26] in the **Authenticate** and **Verify** algorithms. However, it is slightly higher than JXZL [24] because of one time consuming pairing operation. Additionally, note that the schemes JXLZ [23] and JXZL [24] cannot be leveraged for batch verification. Thus at receipt of n authenticators both JXLZ [23] and JXZL [24] needs to run verification for each authenticator one after the other. On the contrary, our CLIBDA scheme can accelerate such a process of verification due its support for batch verification. Although JCYZ [26] also supports batch verification, its computational cost for batch verification is higher than that of our scheme. Schemes in JXLZ [23], JXZL [24] thrives on CLC and are not suitable for heterogeneous systems as in the case of pervasive computing systems. However, like our CLIBDA scheme, JCYZ [26] is heterogeneous and again supports batch verification but at a relatively higher computational cost. The property-wise difference between our scheme and JCYZ [26] is that, while the senders in both our CLIBDA and JCYZ [26] are assumed to be in the same cryptographic environment, the receivers are in IBC and PKI environments respectively. Obviously, the communication overheads of JXLZ [23], JCYZ [26] and our CLIBDA scheme are somewhat high for the reason of the G_2 element (such as W in the case of our scheme) that is meant to be transmitted to the receiver. Following after [26] and the standard in [30], we evaluate the four schemes on an MNT curve of an embedding degree of 6 and additionally 160 bits q on an 32 bit Intel Pentium IV 3.0 GHz PC. The average timing for running pairing computation is 4.5 ms whereas the average time for running point multiplication is also 0.6 ms. Accordingly, Fig. 2 indicates the relationship between the timings for verification against the number of authenticators in these four schemes. JXLZ [23] requires $1P + 2M$ (5.7ms) to run verification for an authenticator and $1nP + 2nM$ (5.7n ms) to run verification

for n authenticators. JXZL [24] requires $6M$ (3.6 ms) to run verification for an authenticator and $6nM$ ($3.6n$ ms) to run verification for n authenticators. JCYZ [26] requires $1P + 2M$ (5.7 ms) to run verification for an authenticator and $1P + (2n - 1)M$ ($1.2n + 3.9$ ms) to run verification for n authenticators. Our CLIBDA scheme requires $1P + 1M$ (5.1 ms) to run verification for an authenticator and $1P + (n - 1)M$ ($0.6n + 3.9$ ms) to run verification for n authenticators. For $n = 100$, the CLIBDA scheme is $\frac{570-63.9}{570} = 88.7\%$ faster than JXLZ [23], $\frac{360-63.9}{360} = 88.3\%$ faster than JXZL [24] and additionally $\frac{123.9-63.9}{123.9} = 48.4\%$ faster than JCYZ [26]. Hence, our scheme is practically fitting for applications in pervasive computing environments.

5.1 Application of CLIBDA

Here we give an example of how CLIBDA scheme can be leveraged for online secure negotiation. Suppose Bob is identified as customer in the IBC environment. One day, Bob decides to buy some goods online and realizes that Alice's goods are pretty good. However, Alice, the merchant runs on the CLC security technology. In an instance like this, a homogeneous deniable authentication scheme can not be leveraged. However, a heterogeneous such as our CLIBDA scheme fits well. Alice may send a price offer m with the authenticator σ to Bob. It is the desire of Alice to assure that Bob is prevented from showing this offer to

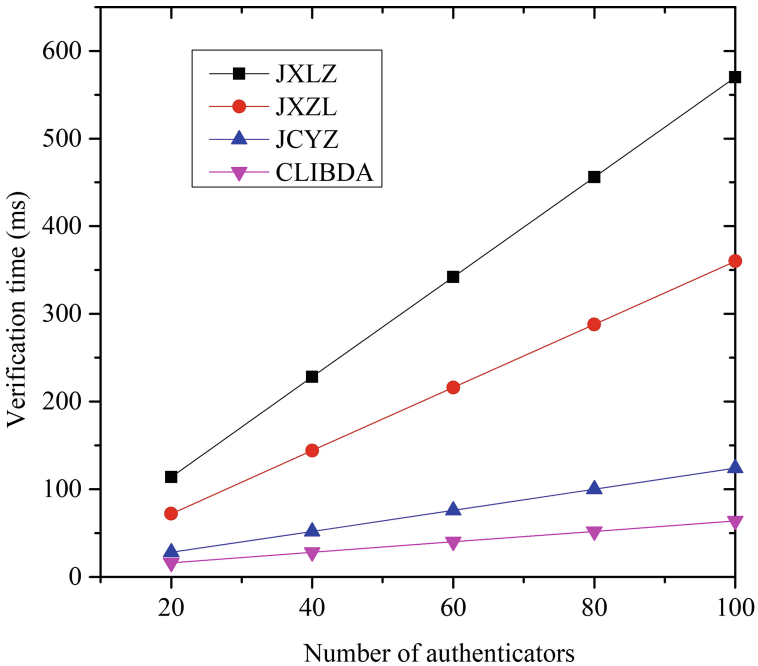


Fig. 2. Number of authenticators against verification time.

a third party. This could help Alice to elicit for a better price from Bob or any future customer. Bob is able to verify that the price offer comes from Alice, but he cannot prove to the third party that Alice gave that price offer. Moreover, the third party cannot also judge whether or not the offer comes from Alice because Bob the customer can actively generate the same m and then authenticator σ .

6 Conclusion

In this paper, we proposed a new heterogeneous deniable authentication scheme called CLIBDA for pervasive computing environments utilizing bilinear pairings. The proposed CLIBDA protocol permits a sender in certificateless cryptography (CLC) setting to transmit a message securely to a receiver in an identity based cryptography (IBC) setting. Detailed security analysis shows that the CLIBDA scheme is secure in the random oracle model (ROM) under the bilinear Diffie–Hellman assumption. Additionally, CLIBDA supports batch verification which is necessary for the speed up of the verification of authenticators. This characteristic makes the CLIBDA scheme suitable in pervasive computing environment applications such as online secure negotiation.

References

1. Alomair, B., Poovendran, R.: Efficient authentication for mobile and pervasive computing. *IEEE Trans. Mob. Comput.* **13**(3), 469–481 (2014)
2. Bettini, C., Riboni, D.: Privacy protection in pervasive systems: state of the art and technical challenges. *Pervasive Mob. Comput.* **17**(1), 159–174 (2015)
3. Ren, K., Lou, W., Kim, K., Deng, R.: A novel privacy preserving authentication and access control scheme for pervasive computing environments. *IEEE Trans. Veh. Technol.* **55**(4), 1373–1384 (2006)
4. Long, M., Wu, C.H.: Energy-efficient and intrusion resilient authentication for ubiquitous access to factory floor information. *IEEE Trans. Ind. Inform.* **2**(1), 40–47 (2006)
5. Yao, L., Wang, L., Kong, X., Wu, G., Xia, F.: An inter-domain authentication scheme for pervasive computing environment. *Comput. Math. Appl.* **60**(2), 234–244 (2010)
6. Tan, Z.: A lightweight conditional privacy-preserving authentication and access control scheme for pervasive computing environments. *J. Netw. Comput. Appl.* **35**(6), 1839–1846 (2012)
7. Park, J.H.: An authentication protocol offering service anonymity of mobile device in ubiquitous environment. *J. Supercomput.* **62**(1), 105–117 (2012)
8. Mayrhofer, R., Fuß, J., Ion, I.: UACAP: a unified auxiliary channel authentication protocol. *IEEE Trans. Mob. Comput.* **12**(4), 710–721 (2013)
9. Wu, Z.Y., Wu, J.C., Lin, S.C., Wang, C.: An electronic voting mechanism for fighting bribery and coercion. *J. Netw. Comput. Appl.* **40**(1), 139–150 (2014)
10. Aumann, Y., Rabin, M.O.: Authentication, enhanced security and error correcting codes. In: Krawczyk, H. (ed.) *CRYPTO 1998*. LNCS, vol. 1462, pp. 299–303. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0055736>

11. Harn, L., Ren, J.: Design of fully deniable authentication service for e-mail applications. *IEEE Commun. Lett.* **12**(3), 219–221 (2008)
12. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) *CRYPTO 1984*. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
13. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Lai, 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
14. Wang, B., Song, Z.: A non-interactive deniable authentication scheme based on designated verifier proofs. *Inf. Sci.* **179**(6), 858–865 (2009)
15. Di Raimondo, M., Gennaro, R.: New approaches for deniable authentication. In: 12th ACM Conference on Computer and Communications Security, pp. 112–121. ACM, Maryland (2005)
16. Tian, H., Chen, X., Jiang, Z.: Non-interactive deniable authentication protocols. In: Wu, C.-K., Yung, M., Lin, D. (eds.) *Inscrypt 2011*. LNCS, vol. 7537, pp. 142–159. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34704-7_12
17. Li, F., Takagi, T.: Cryptanalysis and improvement of robust deniable authentication protocol. *Wirel. Pers. Commun.* **69**(4), 1391–1398 (2013)
18. Gambs, S., Onete, C., Robert, J.: Prover anonymous and deniable distancebounding authentication. In: 9th ACM Symposium on Information Computer and Communications Security, Kyoto, pp. 501–506. ACM (2014)
19. Zeng, S., Chen, Y., Tan, S., He, M.: Concurrently deniable ring authentication and its application to LBS in VANETs. *Peer-to-Peer Netw. Appl.* **10**(4), 844–856 (2017)
20. Lu, R., Cao, Z., Wang, S., Bao, H.: A new ID-based deniable authentication protocol. *Informatica* **18**(1), 67–78 (2007)
21. Li, F., Xiong, P., Jin, C.: Identity-based deniable authentication for ad hoc networks. *Computing* **96**(9), 843–853 (2014)
22. Yao, A., Zhao, Y.: Privacy-preserving authenticated key-exchange over Internet. *IEEE Trans. Inf. Forensics Secur.* **9**(1), 125–140 (2014)
23. Jin, C., Xu, C., Li, F., Zhang, X.: A novel certificateless deniable authentication protocol. *Int. J. Comput. Appl.* **37**(3–4), 181–192 (2015)
24. Jin, C., Xu, C., Zhang, X., Li, F.: An efficient certificateless deniable authentication protocol without pairings. *Int. J. Electron. Secur. Digit. Forensics* **7**(2), 179–196 (2015)
25. Li, F., Hong, J., Omala, A.: Practical deniable authentication for pervasive computing environments. *Wirel. Netw.* **24**(1), 139–149 (2018)
26. Jin, C., Chen, G., Yu, C., Zhao, J.: Heterogeneous deniable authentication for e-voting systems. In: Li, F., Takagi, T., Xu, C., Zhang, X. (eds.) *FCS 2018*. CCIS, vol. 879, pp. 41–54. Springer, Singapore (2018). https://doi.org/10.1007/978-981-13-3095-7_4
27. Choon, J.C., Hee Cheon, J.: An identity-based signature from gap Diffie-Hellman groups. In: Desmedt, Y.G. (ed.) *PKC 2003*. LNCS, vol. 2567, pp. 18–30. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36288-6_2
28. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) *CRYPTO 2001*. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
29. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. *J. Cryptol.* **13**(3), 361–396 (2000)
30. Scott, M.: Efficient implementation of cryptographic pairings (2007). <http://www.pairing-conference.org/2007/invited/Scottslide.pdf>



Leveled Lattice-Based Linearly Homomorphic Signature Scheme in the Standard Model for Network Coding

Fenghe Wang¹(✉), Shaoquan Shi², and Chunxiao Wang¹

¹ School of Science, Shandong Jianzhu University, Jinan, China
fenghe2166@163.com

² School of Computer, Shandong Jianzhu University, Jinan, China

Abstract. Linearly homomorphic signature scheme is an important cryptographic primitive which can be used to against the pollution attacks in network coding. To achieve the security protection for network coding even in quantum environment, an efficient lattice-based linearly homomorphic signature scheme in the standard model is proposed in this paper. Unlike the known lattice-based scheme in the standard model, in our construction, lattice-based delegation algorithm is not needed to achieve the standard security. Hence, all the messages are signed over the same lattice in the proposed scheme. Hence, the public key of the proposed scheme only consists as a group of vectors compared with that a group of public and random matrices are necessary in known construction used lattice-based delegation tool. As a result, the public key size of the proposed scheme is shorter than that of the known lattice-based schemes (standard model). Moreover, the proposed scheme also shares advantage about the signature length. Based on the hardness of the standard short integer solution problem, we prove that the proposed scheme is adaptively unforgeable against the type 1 and type 2 adversaries in the standard model. We also shown that the proposed scheme satisfies the weakly context hiding property.

Keywords: Linearly homomorphic signature · Standard model · Lattice · Short integer solution · Pre-image sampling function

1 Introduction

In network coding[10], the intermediate nodes acts as data packets in transit which shares some advantages especially in wireless and/or ad-hoc networks. While the security should be pay more attentions when the network coding is used in practical applications. The mayor security concern is to provide protection against the pollution attacks by the malicious nodes. Because the intermediate nodes in network coding can work as an information processor, while the standard signature can not depend the computations of the signatures. Then, the standard signature scheme can not be used directly to proposed the

authentication. Linearly homomorphic signature(LHS) which depends the computations of the signatures [15], can provide the protection against the pollution attack by the malicious node in the network coding.

In fact, one can use LHS scheme to generate easily the signature on any vector \mathbf{v} in the \mathbb{F}_p -linear span of $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ by the linear homomorphic property where $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ are defined over a finite field \mathbb{F}_p with their signatures has been generated by a LHS algorithm. It is clearly that LHS can be used as a cryptographic solution to guarantee the correctness and the authenticity of the delegated computation which is important to both the network coding and cloud computing [16].

Most previous results on LHS are built based on the number theorem assumption such as the discrete logarithm problem or the RSA problem etc. [1, 4, 8, 11, 12]. Although yield very elegant constructions, these schemes must be designed over \mathbb{F}_p for some large p to sure the hardness of the RSA problem or discrete logarithm problem. First LHS scheme on \mathbb{F}_2 was presented by Boneh and Freeman [5] whose security is based on k -SIS problem (k -small integer solution problem). Subsequently, the same authors proposed the lattice-based homomorphic signature that can compute constant degree polynomials on signed data. The security of this scheme is based on the SIS problem over the ideal lattice [6]. Wang etc. proposed an efficient scheme over \mathbb{F}_2 whose security is directly based on the standard SIS problem [19]. More details about the progress of the homomorphic signature can be found in [8]. Note that such schemes [5, 6, 19] are all designed over lattice whose security are proven in the random oracles model. Nevertheless the ideal random oracles are hard to achieve in the present world. Chen et.al proposes a LHS scheme over small field in the standard model by using lattice-based delegation technology [9]. While the public key sizes of that schemes are too large to be used in network coding. In fact, the public key of this scheme consists as $2r + 1$ matrices over $\mathbb{Z}_q^{n \times m}$ and k vectors over \mathbb{Z}_q^n . As a result, how to short the public key size of the lattice-based LHS scheme in the standard model is an interesting topic which should be pay more attentions.

Our Work. This paper builds an efficient LHS scheme in the standard model whose security is based on the standard SIS problem. Different from the construction in [9], lattice-based delegation technology [7] are not used in this paper. Since then we do not need to use some public matrices as the public key which are used to generate a new lattice. In fact, we only use a group of vectors to encode the message before inputting the sign algorithm. More precisely, to achieve the standard model security, we firstly encode the message under some random vectors. And then, the presample function [13] is used to sign the encoded vector. We should note that the coding technology used in this paper would help us to finish the signature simulation in the security proof phase. Since the message is encoded linearly, the linearly homomorphic property of the proposed scheme is easily hold. As a result, the public key size of our construction is smaller than that of the scheme in [9]. At last, we show that the proposed scheme is efficient with respect to the public key size and signature length etc. In fact, it even

more efficient than that of the standard lattice-based signature schemes in the standard model.

Relate Work. Lattice, known as a subgroup of a vector space, has been found many applications in cryptography field in recent years. Moreover, the lattice-based cryptography is considered still secure even under quantum attacks. Hence, Lattice-based cryptography has gain more and more attentions in recent years. Many important results have been achieved in lattice-based signature fields, such as lattice-based signature schemes [7, 13, 17], linearly homomorphic signature schemes [5, 9, 19] and fully homomorphic signature schemes [2, 6, 14, 20]. The leveled fully homomorphic signature schemes in the standard model were present in [14, 20] which all can evaluate arbitrary circuit (not only linear function) over the signed data. If the adversary is ask to sure the message whose signature will be forged before the security game runs, the schemes in [14] is provable secure in the standard model. The scheme in [20] by Boyen et.al achieves the adaptive security.

2 Primitives

2.1 Notations

Bold upper-case and the bold lower-case letters denote matrices and vectors in column form respectively. If $\|\cdot\|$ is the Euclidean norm, the norm of the longest column is defined to be the matrix norm. The Gram-Schmidt orthogonalized matrix is written by $\mathbf{\bar{T}}$. $poly(n)$ is an unspecified function $f(n) = O(n^c)$ for a constant c . Function $g(n)$ is negligible if $g(n) = 1/poly(n)$. We see a function $g(n) = \omega(f(n))$ if it grows faster than $cf(n)$ for any constant c . D_α denotes the Gaussian distribution over \mathbb{R} with parameter α .

2.2 Lattice

Defined

$$A = \{\mathbf{Bc} = \sum_{i \in [n]} c_i \mathbf{b}_i, |c_i \in \mathbb{Z}\},$$

be a lattice generated by a basis \mathbf{B} which $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$ which are linearly independent vectors. Moreover, \mathbf{B} is called trapdoor basis if all \mathbf{b}_i are with small Euclidean norms.

For a prime number q and $\mathbf{y} \in \mathbb{Z}_q^n$, two special integer lattice are usually considered in the design of lattice-based cryptographic scheme,

$$\begin{aligned} A_q^\perp(\mathbf{A}) &= \{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{Ax} = 0(\text{mod } q)\}, \\ A_q^\mathbf{y}(\mathbf{A}) &= \{\mathbf{x} \in \mathbb{Z}_q^m, \mathbf{Ax} = \mathbf{y}(\text{mod } q)\}. \end{aligned}$$

The SIS problem defined over $A_q^\perp(\mathbf{A})$ is widely used to sure the security of the lattice-based signature scheme. Let (n, m, q) be parameters.

Definition 1. *The SIS problem is defined as follows: given a real β and $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find a nonzero vector $\mathbf{e} \in \mathbb{Z}_q^m$ satisfying $\mathbf{Ae} = 0(\text{mod } q)$ and $\|\mathbf{e}\| \leq \beta$.*

2.3 Discrete Gaussian Distribution

The discrete Gaussian distribution on lattice $\Lambda_q^\perp(\mathbf{A})$ ($\mathbf{A} \in \mathbb{Z}_q^{n \times m}$) is defined by a “conditional” distribution

$$D_{\Lambda_q^\perp(\mathbf{A}), \sigma, \mathbf{c}}(x) = \frac{\rho_{\sigma, \mathbf{c}}(x)}{\rho_{\sigma, \mathbf{c}}(\Lambda_q^\perp(\mathbf{A}))}.$$

Smoothing parameter $\eta_\epsilon(\Lambda)$ is defined to be the smallest positive σ satisfying $\rho_{1/\sigma}(\Lambda^* \setminus \{0\}) \leq \epsilon$ where Λ is a lattice and $\epsilon > 0$ is a positive real[18]. It has been proven that when $\sigma > \eta_\epsilon(\Lambda)$, every coset of Λ has roughly equal mass. Moreover, for almost ϵ and random lattice $\Lambda_q^\perp(\mathbf{A})$, $\eta_\epsilon(\Lambda_q^\perp(\mathbf{A})) > \omega(\sqrt{\log m})$ would hold with overwhelm probability.

There are several lemmas which are important for the design in this paper.

Lemma 1. [3] *The Trapdoor Sampling Algorithm outputs $(\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{S})$ in probabilistic polynomial time where \mathbf{A} is statistically close to the uniform distribution over $\mathbb{Z}_q^{n \times m}$ and $\|\mathbf{S}\| \leq O(n \log q)$. Furthermore, \mathbf{S} can be efficiently converted to be a trapdoor basis of the lattice $\Lambda_q^\perp(\mathbf{A})$.*

Lemma 1 shows that $\Lambda_q^\perp(\mathbf{A})$ with its trapdoor basis can be generated in polynomial time [3].

Lemma 2. $\mathbf{t}_i \in \mathbb{Z}^m$ and \mathbf{x}_i are mutually independent random variables sampled from a Gaussian distribution $D_{\mathbf{t}_i + \Lambda, \sigma}$ over $\mathbf{t}_i + \Lambda$ for $i = 1, 2, \dots, k$ in which Λ is a lattice and $\sigma \in \mathbb{R}$ is a parameter. Let $\mathbf{c} = (c_1, \dots, c_k) \in \mathbb{Z}^k$ and $g = \gcd(c_1, \dots, c_k)$, $\mathbf{t} = \sum_{i=1}^k c_i \mathbf{t}_i$. If $\sigma > \|\mathbf{c}\| \eta_\epsilon(\Lambda)$ for some negligible number ϵ , then $\mathbf{z} = \sum_{i=1}^k c_i \mathbf{x}_i$ statistically closes to $D_{\mathbf{t} + g\Lambda, \|\mathbf{c}\|\sigma}$.

Lemma 2 [5] will help us to prove the weakly content privacy of the proposed scheme. The next lemma shows that it is possible to sample a Gaussian vector with short norm by using a trapdoor basis of lattice.

Lemma 3. [13]. *There is algorithm PreSampleD outputs vector \mathbf{e} with a distribution that is statistically close to $D_{\Lambda, s, \mathbf{c}}$, where inputs a basis \mathbf{B} of Λ , $s > \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log n})$ and $\mathbf{c} \in \mathbb{R}^n$.*

Lemma 4. [13] *Given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ whose columns of generate \mathbb{Z}_q^n , $\epsilon \in (0, 1)$ and $s > \eta_\epsilon(\Lambda_q^\perp(\mathbf{A}))$, if $\mathbf{e} \sim D_{\mathbb{Z}^m, s}$, then the distribution of the syndrome $\mathbf{u} = \mathbf{A}\mathbf{e}(\text{mod } q)$ is within statistical distance 2ϵ of the uniform distribution.*

2.4 Linearly Homomorphic Signature

Definition 2. *Let $M \in V \subset \mathbb{F}_2^n$ denote a message. The LHS scheme is defined as follows:*

Kg. *Given a parameter 1^n , this algorithm outputs $(pk; sk)$ as the public/private key of the signer.*

Sign. It generates a signature $\mathbf{e} = \text{Sign}(\mathbf{M}, id, sk)$ where $id \in \{0, 1\}^*$ is an identifier of the message set V .

Vrf. Inputting $(\mathbf{M}, \mathbf{e}, id, pk)$, it outputs $b = 1$ when \mathbf{e} is a valid signature of \mathbf{M} , otherwise it outputs $b = 0$.

Combine. Given signatures $(\mathbf{e}_i, \mathbf{M}_i)$, an identifier id , public key pk and coefficients $a_i \in \{0, 1\}$ for $i \leq L$, it outputs the signature \mathbf{e} for the message $\mathbf{M} = \sum a_i \mathbf{M}_i \pmod{2}$.

The security properties of a LHS scheme are Correctness, Privacy and Unforgeability.

Correctness. A signature properly formed by the Sign algorithm or the Combine algorithm can be accepted by the Vrf algorithm.

Privacy. We only consider the weakly context hiding[5] which means that the signature \mathbf{v} derived from signatures \mathbf{v}_i does not leak the information about \mathbf{v}_i . Other strong context hiding can be found in [1].

Definition 3. The weakly context hiding of the LHS scheme can be defined by the following security game.

Setup. The challenger generates and sends (pk, sk) to the adversary by the Kg algorithm.

Challenge

The Adversary. Outputs two linear subspaces V_0, V_1 represented as k -tuples of vectors $(\mathbf{v}_1^{(b)}, \dots, \mathbf{v}_k^{(b)})$ for $b = 0, 1$ respectively. The function f_1, \dots, f_s satisfy $f_i(\mathbf{v}_1^{(0)}, \dots, \mathbf{v}_k^{(0)}) = f_i(\mathbf{v}_1^{(1)}, \dots, \mathbf{v}_k^{(1)})$ for all $i = 1, 2, \dots, s$.

The Challenger 1. Generates a random bit $b \in \{0, 1\}$, an identifier id and signs the vector space V_b .

2. Derives signatures \mathbf{e} on $f_i(\mathbf{v}_1^{(b)}, \dots, \mathbf{v}_k^{(b)})$ in which the function can be output adaptively after V_0, V_1 are output. Sends \mathbf{e} to the adversary.

Outputs. The adversary guesses a bit b' , if $b = b'$, the adversary wins the game. If the advantage of any PPT adversary in above game is negligible, we call a LHS scheme satisfies weakly context hiding.

Unforgeability. The unforgeability of the LHS is defined as follows:

Definition 4. A LHS scheme satisfies unforgeability under the chosen message attack if any PPT adversary's advantage is negligible in the following game.

Setup. The challenger generates (pk, sk) by the Kg algorithm. It also sends pk to the adversary.

Sign Queries. The adversary can query the signature of $\mathbf{v}_{i1}, \dots, \mathbf{v}_{ik}$ which is a basis of a k -dimensional subspaces V_i .

For V_i and $j = 1, 2, \dots, k$, the sign oracle chooses $id^{(i)}$ uniformly and gives $id^{(i)}, \mathbf{e}_{ij}$ to the adversary.

Output: The adversary outputs $(id^* \in \{0, 1\}^n M^*, S)$ where S is a signature of a new message M^* .

Two types adversary both are considered in this game.

(1)(type1) $id^* \neq id^{(i)}$ for all i ;

(2)(type2) $id^* = id^{(i)}$ for some i , $M^* \notin V_i$.

If (id^*, M^*, S) can be accepted by the Vrf algorithm then the adversary wins the game.

3 Linearly Homomorphic Signature Scheme in the Standard Model

Given a prime number n , and a constant $c > 0$, then $q \geq \beta\omega(\log n)$ for $\beta = poly(n)$, $m \geq cn \log q$, $\tilde{L} \geq O(\sqrt{n \log q})$ and $\sigma = \tilde{L}\omega(\sqrt{\log n})$. The maximal number of signatures which can be combined in the proposed LHS scheme is denoted by L . Both the identifier of file and message belong to \mathbb{Z}_2^k where $k < \sqrt{\tilde{L}}$. Let the sender be Alice and the recipient verifier Bob.

Kg. Alice generates $\mathbf{A} \in \mathbb{Z}_{2q}^{n \times m}$ and its trapdoor basis $\mathbf{T} \in \mathbb{Z}_{2q}^{m \times m}$ (Lemma 1). Randomly chooses k vectors over \mathbb{Z}_{2q}^m denoted respectively by $\mathbf{c}_1, \dots, \mathbf{c}_k$.

Then the public key of Alice is $(\mathbf{A}, \mathbf{c}_1, \dots, \mathbf{c}_k)$ and private key \mathbf{T} .

Sign. Given a message vector $\mathbf{v}_i = (v_{i1}, \dots, v_{ik}) \in \{0, 1\}^k \subset V$ and an identifier $\mathbf{id} = (id_1, id_2, \dots, id_k) \in \{0, 1\}^k$ of the subspace V , Alice generates the signature of \mathbf{v}_i as follows:

1. Computes $\mathbf{v}'_i \mathbf{v}'_i = \sum_{j=1}^k q(-1)^{id_j} v_{ij} \mathbf{c}_j \pmod{2q}$.
2. Generates a preimage of \mathbf{v}'_i :

$$\mathbf{e}_i \leftarrow PreSampleD(\mathbf{A}, \mathbf{T}, \sigma, \mathbf{v}'_i).$$

Hence, $(\mathbf{e}_i, \mathbf{id})$ is the signature of \mathbf{v}_i .

Note that the sign algorithm at most to sign l linearly independence message vectors of a l -dimensional message subspace in the proposed scheme.

Vrf. To verify the signature $(\mathbf{e}_i, \mathbf{id})$ of the message \mathbf{v}_i , Bob does as follows:

1. Computes $\mathbf{v}'_i = \sum_{j=1}^k q(-1)^{id_j} v_{ij} \mathbf{c}_j \pmod{2q}$.
2. Accepts $(\mathbf{e}_i, \mathbf{id})$ if and only if:

$$(a). \mathbf{A}\mathbf{e}_i = \mathbf{v}'_i \pmod{2q}; \quad (b). \|\mathbf{e}_i\| \leq L\sigma\sqrt{m}.$$

Combine. Given signatures $(\mathbf{id}, \mathbf{e}_i)$ of message $\mathbf{v}_i \in \{0, 1\}^k$ and coefficients $\alpha_i \in \{0, 1\}$ for $i = 1, 2, \dots, l$ and $l \leq L$, the signature of message $\sum_{i=1}^l \alpha_i \mathbf{v}_i \pmod{2}$

is given by $\sum_{i=1}^l \alpha_i \mathbf{e}_i \pmod{2q}$.

4 Analysis of the Proposed LHS scheme

4.1 Correctness

Proof. If \mathbf{e}_i is directly generated by the sign algorithm, it is an output of the PSF algorithm. According to literature [13], we know that $\mathbf{A}\mathbf{e}_i = \mathbf{v}(\bmod 2q)$ and $\|\mathbf{e}_i\| \leq \sigma\sqrt{m}$ hold. Hence it can be accepted by the Vrf algorithm.

If $(\mathbf{id}, \mathbf{e})$ is an output of the Combine algorithm, we show that it also can be accepted by the Vrf algorithm.

Considered $(\mathbf{id}, \mathbf{e})$ generated by $(\mathbf{id}, \mathbf{e}_1)$ and $(\mathbf{id}, \mathbf{e}_2)$ for simply. Let $\mathbf{v}_1 = (v_{11}, v_{12}, \dots, v_{1k})$ and $\mathbf{v}_2 = (v_{21}, v_{22}, \dots, v_{2k})$ be the the message. Hence $(\mathbf{id}, \mathbf{e} = \mathbf{e}_1 \pm \mathbf{e}_2)$ is the signature of $\mathbf{v} = \mathbf{v}_1 \pm \mathbf{v}_2 = (v_1, v_2, \dots, v_k)$. In fact,

$$\begin{aligned}\mathbf{A}\mathbf{e}_1 &= \sum_{j=1}^k q(-1)^{id_j} v_{1j} \mathbf{c}_j(\bmod 2q), \\ \mathbf{A}\mathbf{e}_2 &= \sum_{j=1}^k q(-1)^{id_j} v_{2j} \mathbf{c}_j(\bmod 2q).\end{aligned}$$

Then

$$\begin{aligned}\mathbf{A}(\mathbf{e}_1 \pm \mathbf{e}_2) &= \sum_{j=1}^k q(-1)^{id_j} (v_{1j} \pm v_{2j}) \mathbf{c}_j(\bmod 2q) \\ &= \sum_{j=1}^k q(-1)^{id_j} (v_j) \mathbf{c}_j(\bmod 2q).\end{aligned}$$

On the other hand, $\|\mathbf{e}_1 \pm \mathbf{e}_2\| \leq Ls\sqrt{m}$ holds.

So that $(\mathbf{id}, \mathbf{e})$ is the signature of the message vector $\mathbf{v} = (\mathbf{v}_1 \pm \mathbf{v}_2)(\bmod 2)$.

The general case can be proven by the same way.

4.2 Security

Theorem 1. *If there is an adversary can win the unforgeability game of the LHS scheme with a probability ε , there is a challenger can solve the SIS problem with a probability approaching $\varepsilon/(q-1)$.*

Proof. Suppose there exists a PPT adversary \mathcal{A} gaining an advantage ε for winning the unforgeability game, a PPT challenger \mathcal{C} can be constructed to solve the SIS problem with advantage $2\varepsilon/(q-1)$.

Suppose that the challenger \mathcal{C} wants to solve an SIS instance $(\mathbf{A}_0 \in \mathbb{Z}_{2q}^{n \times m}, q, n, \sigma, L)$. That is, it hopes to find a vector \mathbf{e} satisfying $\|\mathbf{e}\| \leq 2L\sigma\sqrt{m}$ and $\mathbf{A}_0\mathbf{e} = \mathbf{0}(\bmod 2q)$. \mathcal{C} does as follows:

- (1) Sets $\mathbf{A} = q\mathbf{A}_0(\text{mod } 2q)$;
- (2) Randomly chooses $\bar{\mathbf{e}}_i$ for $i = 1, 2, \dots, k$ according to the distribution D_s^m where $s = \frac{\tilde{L}}{\sqrt{k}}\omega(\sqrt{\log n})$ and ensures these vectors are linearly independent. According to [13], at most chosen k^2 vectors, we can get and ensure $\bar{\mathbf{e}}_i$ for $i = 1, 2, \dots, k$ are linearly independent.
- (3) Sets $\mathbf{c}_i = \mathbf{A}_0\bar{\mathbf{e}}_i(\text{mod } 2q)$.

\mathcal{C} sends $\{\mathbf{A}, \{\mathbf{c}_i\}_{i=1}^k\}$ to \mathcal{A} as the public key. To finish the security game, \mathcal{C} keeps a list to store the answers to the signature oracle.

Sign Query. When the challenger needs to answer a sequence of sign queries, it firstly checks the freshness of this query by the list. If the message subspace V_i had been queried then the same answers are returned. For a fresh subspace, if V_i is represented by k vectors pair $\mathbf{v}_i \in \mathbb{Z}_2^m$, \mathcal{C} chooses an identifier $\mathbf{id}^{(i)} \in \{0, 1\}^k$ and computes $\mathbf{e}_i = \sum_{t=1}^k (-1)^{\mathbf{id}_t^{(i)}} v_{it} \bar{\mathbf{e}}_t$. Then, according to Lemma 2, \mathbf{e}_i are distributed to Gaussian distribution D_σ^m with an overwhelm probability where $\sigma = \frac{\tilde{L}}{\sqrt{k}}\omega(\sqrt{\log n}) \times \sqrt{k} = \tilde{L}\omega(\sqrt{\log n})$. \mathcal{C} stores $(\mathbf{id}^{(i)}, \mathbf{v}_i, \mathbf{e}_i)$ to the list L . \mathcal{C} outputs $(\mathbf{id}^{(i)}, \mathbf{v}_i, \mathbf{e}_i)$. Then the adversary can compute $\mathbf{v}'_i = \sum_{j=1}^k q(-1)^{\mathbf{id}_j} v_{ij} \mathbf{c}_j(\text{mod } 2q)$ and check that $\|\mathbf{e}_i\| \leq L\sigma\sqrt{m}$ and

$$\mathbf{A}\mathbf{e}_i = \mathbf{A} \sum_{t=1}^k (-1)^{\mathbf{id}_t^{(i)}} v_{it} \bar{\mathbf{e}}_t = \sum_{t=1}^k (-1)^{\mathbf{id}_t^{(i)}} v_{it} q \mathbf{A}_0 \bar{\mathbf{e}}_t = \sum_{t=1}^k (-1)^{\mathbf{id}_t^{(i)}} v_{it} q \mathbf{c}_t = \mathbf{v}'_i$$

After all the queries are finished, \mathcal{A} forges a signature $(\mathbf{e}^*, \mathbf{id}^*)$ of the message \mathbf{v}^* with the probability ε .

- (1) For a type 1 adversary, \mathbf{id}^* never be queried. The challenger computes $\mathbf{e} = \sum_{j=1}^k (-1)^{\mathbf{id}_j^*} v_j^* \bar{\mathbf{e}}_j$.
- (2) For a type 2 adversary, \mathbf{id}^* has be queried while the message $\mathbf{v}^* \notin V_i$. The challenger finds $(\mathbf{id}^{(i)}, \{\mathbf{e}_i\}_{i=1}^k)$ from the list L . Set $\mathbf{e} = \sum_{j=1}^k (-1)^{\mathbf{id}_j^*} v_j^* \bar{\mathbf{e}}_j$ as the other signature on \mathbf{v}^* .

$$\text{Then } \mathbf{A}\mathbf{e} = \sum_{j=1}^k q(-1)^{\mathbf{id}_j^*} v_j \mathbf{c}_j(\text{mod } 2q).$$

$$\text{Hence } \mathbf{A}(\mathbf{e}^* - \mathbf{e}) = 0(\text{mod } 2q).$$

$$\text{That is } q\mathbf{A}_0(\mathbf{e}^* - \mathbf{e}) = 0(\text{mod } 2q).$$

It means that $\mathbf{A}_0(\mathbf{e}^* - \mathbf{e}) = 2l\mathbf{c}(\text{mod } 2q)$ where $0 \leq l < q$ and $\mathbf{c} \in \mathbb{Z}_{2q}^n$.

Since \mathbf{A}_0 is uniform and \mathbf{e}^*, \mathbf{e} are gaussian vectors, according to [13] and Lemma 2, $\mathbf{A}_0(\mathbf{e}^* - \mathbf{e})$ is a uniform vector with an overwhelm probability. Hence the coefficient $l = 0, 1, \dots, (q-1)$ with equal probability. On the other hand, $\|\mathbf{e}^* - \mathbf{e}\| \leq 2Ls\sqrt{m}$ and $\mathbf{e}^* \neq \mathbf{e}$ with probability $1 - 2^{-\omega(\log n)}$ [13]. It is clear that \mathcal{C} solves the SIS problem if $k = 0$ holds with a probability $\varepsilon(1 - 2^{-\omega \log n}) / (q-1)$.

As a result, \mathcal{C} would solve the SIS problem with a probability $\frac{1}{q-1}(1 - 2^{-\omega \log n})\varepsilon$.

Theorem 2. *The proposed LHS scheme satisfies the weakly context hiding property.*

Proof. Let $V_b = \text{span}\{\mathbf{v}_1^b, \dots, \mathbf{v}_k^b\}$ for $b = 0, 1$. For $j = 1, 2, \dots, k$, $\mathbf{e}_j^{(0)}$ and $\mathbf{e}_j^{(1)}$ are the signatures on the basis message vectors of V_0 and V_1 respectively. Hence $\mathbf{e}_j^{(0)}$ and $\mathbf{e}_j^{(1)}$ are statistically close to the Gaussian distribution [13].

For a bit $b \in \{0, 1\}$ chosen by the challenger, $\mathbf{e}^{(b)}$ is the combine signature which is derived from $\mathbf{e}_j^{(b)}$ where $j = 1, 2, \dots, k$.

If $b = 0$, $\mathbf{e}^{(0)}$ is derived from $\mathbf{e}_j^{(0)}$ under some linearly functions f_i for $i = 1, \dots, s$. From Lemma 2 we know that the distribution of $\mathbf{e}^{(0)}$ is statistically close to the Gaussian distribution depending on $(\Lambda, \sigma, f(V_0), f)$ where f belongs to the functions set f_1, \dots, f_s .

If $b = 1$, the same fact holds. Moreover, $f_i(V_0) = f_i(V_1)$ for $i = 1, 2, \dots, s$.

Therefore the distributions of $\mathbf{e}^{(0)}$ and $\mathbf{e}^{(1)}$ are statistically closed. Consequently, no PPT adversary can win the privacy game.

4.3 Efficiency

We compared the proposed scheme with the known LHS scheme over lattice in the standard model [9] by the following table. Table 1 shows that the proposed scheme shares some advantages with respect about the public key size and signature length. More precisely, to easily compare the efficiency, we suppose that the parameters of these schemes are equal. Let (n, m) be the row and column of the lattice matrix respectively. Let k be the length of the tag. Then Table 1 gives the details of the efficiency comparison.

Table 1. Efficiency comparison

Schemes	Public key size (bits)	Signature Length (bits)	Homomorphic
Our scheme	$(mn + k)\log(2q)$	$m\log(2q) + k$	Linear
[9]	$((2k + 1)mn + mk)\log q$	$(k + 1)\log q$	Linear

5 Conclusions

We present an efficient LHS scheme in the standard model over F_2 by using the lattice-based cryptographic tool. The security of the proposed LHS scheme is based on the standard SIS problem. Since we use the data coding technology to replace the lattice-based delegation, we achieve the standard model security without depending on the “grow” of the lattice which means that the public key of the proposed scheme does not consist as a group of the public matrices. Moreover all the messages are signed over the same lattice, hence the signature length of the proposed scheme is also short. As a result, the proposed scheme shares some advantages with respect to public key size and the signature length.

As the further study direction, before wide applicants [21], how to achieve the full homomorphic property of signature scheme in the standard model with the short public key size and the signature length is an interesting topic.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant 61803228, Project of Shandong Province Higher Education Science and Technology Program under grant J18KA361.

References

1. Ahn, D.H., Boneh, D., Camenisch, J., et al.: Computing on authenticated data. *J. Crypt.* **28**(2), 351–395 (2015)
2. Arita, S., Kozaki, S.: A homomorphic signature scheme for quadratic polynomials, in *Smart Computing (SMARTCOMP)*. In: 2017 IEEE International Conference on, IEEE, pp. 1–6 (2017)
3. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. In: *Proceedings of 26th International Symposium on Theoretical Aspects of Computer Science*, vol. 09001, Freiburg, Germany, pp. 75–86 (2009)
4. Boneh, D., Freeman, D.M., Katz, J., et al.: Singing a linear subspace: signature schemes for network coding. In: *Proceedings of PKC 2009, LNCS 5443*, pp. 68–87. Springer-Verlag, Berlin (2009)
5. Boneh, D., Freeman, D.M.: Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) *PKC 2011. LNCS*, vol. 6571, pp. 1–16. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_1
6. Boneh, D., Freeman, D.M.: Homomorphic signatures for polynomial functions. In: Paterson, K.G. (ed.) *EUROCRYPT 2011. LNCS*, vol. 6632, pp. 149–168. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_10
7. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) *EUROCRYPT 2010. LNCS*, vol. 6110, pp. 523–552. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_27
8. Catalano, D., Fiore, D., Nizzardo, L.: Homomorphic signatures with sublinear public keys via asymmetric programmable hash functions. *Des. Codes Cryptogr.* **86**, 2197–2246 (2018)
9. Chen, W., Lei, H., Qi, K.: Lattice-based linearly homomorphic signatures in the standard model. *Theor. Comput. Sci.* **634**, 47–54 (2016)
10. Fragouli, C., Soljanin, E.: Network coding fundamentals. *Found. Trends Netw.* **2**(1), 1–133 (2007)
11. Freeman, D.M.: Improved security for linearly homomorphic signatures: a generic framework. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) *PKC 2012. LNCS*, vol. 7293, pp. 697–714. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30057-8_41
12. Gennaro, R., Katz, J., Krawczyk, H., Rabin, T.: Secure network coding over the integers. In: Nguyen, P.Q., Pointcheval, D. (eds.) *PKC 2010. LNCS*, vol. 6056, pp. 142–160. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_9
13. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing STOC 2008, British Columbia, Canada*, pp. 197–206 (2008)

14. Gorbunov, S., Vaikuntanathan, V., Wichs, D.: (Leveled) fully homomorphic signatures from lattices. In: Proceedings of STOC, pp. 469–477 (2015)
15. Johnson, R., Molnar, D., Song, D., Wagner, D.: Homomorphic signature schemes. In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, pp. 244–262. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45760-7_17
16. Liu, H.W., Cao, W.M.: Public proof of cloud storage from lattice assumption. *Chin. J. Electron.* **23**(1), 186–190 (2014)
17. Lyubashevsky, V., Micciancio, D.: Asymptotically efficient lattice-based digital signatures. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 37–54. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78524-8_3
18. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. In: Proceedings of 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS), Rome, Italy, pp. 372–381 (2004)
19. Wang, F., Hu, Y., Wang, B.: Lattice-based linearly homomorphic signature scheme over binary field. *Sci. China Inf. Sci.* **56**(11), 112108:1–112108:9 (2013)
20. Boyen, X., Fan, X., Shi, E.: Adaptively secure fully homomorphic signatures based on lattices. IACR Cryptology ePrint Archive, 916 (2014)
21. Zheng, Y., Robert, H.D., Vijay, V.: Cryptography and data security in cloud computing. *Inf. Sci.* **387**, 53–55 (2017)



Symmetric Lattice-Based PAKE from Approximate Smooth Projective Hash Function and Reconciliation Mechanism

Zilong Wang^(✉), Honggang Hu^(✉), Mengce Zheng, and Jiehui Nan

Key Laboratory of Electromagnetic Space Information,
Chinese Academy of Sciences, School of Information Science and Technology,
University of Science and Technology of China, Hefei 230027, China
{wz10830,mczheng,ustcnjh}@mail.ustc.edu.cn,
hghu2005@ustc.edu.cn

Abstract. Password-based authenticated key exchange (PAKE) protocols allow two users who share only a short, low-entropy password to establish a consistent cryptographically strong session key. In 2009, Katz and Vaikuntanathan gave the first lattice-base PAKE from approximate smooth projective hash function (ASPHF) which is a variant of smooth projective hash function (SHPF). In 2017, Zhang and Yu introduced a two-round PAKE based on splittable PKEs. An error-correcting code (ECC) was used in these protocols to deal with the errors intrinsically in learning with errors (LWE) assumption, and the protocol is asymmetric as the session key is decided by just one user. In this paper, an error correcting technique called reconciliation mechanism, which was first introduced to construct a key exchange protocol from lattice, is adopted to construct more efficient lattice-based PAKEs with reduced computation complexity and communication complexity. Moreover, the new PAKEs are symmetric.

Keywords: Lattice-based cryptosystem · PAKE · Approximate smooth projective hash function · Reconciliation mechanism

1 Introduction

Cryptography is an important tool for achieving cyberspace security, and key exchange (KE) is one of the most fundamental cryptographic primitives, which can date back to the work of Diffie and Hellman [18]. The original Diffie-Hellman (DH) key exchange protocol is vulnerable to the man-in-the-middle attack. To overcome this issue, Bellare and Rogaway [6] considered a new protocol called authenticated key exchange (AKE) which combines entity authentication with key exchange. AKE allows each party to authenticate the identities of others with the help of some pre-shared information which can be either a high-entropy cryptographic key (such as a secret key from PKI) or a low-entropy string (such as password). Password-based authenticated key exchange (PAKE), which allows

each party to authenticate the identities of others with the help of password, is one of the most widely used protocols as lots of Internet applications are based on PAKE (such as email, e-commerce platforms, social networking services). However, passwords are generated by human beings. It means that passwords are short and easily memorizable, which can be regarded as a string chosen from a small dictionary [9].

The first PAKE protocol, called encrypted key exchange (EKE), was proposed by Bellare and Merritt [7] in 1992. EKE encrypts all the communication data with a block cipher, whose secret key is the password, to guard against the off-line dictionary attacks. After that, a plenty of new PAKE protocols in the random oracle/ideal cipher models have been proposed aiming at higher efficiency by more ingenious designs [5, 12, 30]. In 2001, Katz, Ostrovsky and Yung [24] proposed a practical PAKE (KOY) with BPR security. Inspired by KOY, Gennaro and Lindell [21] gave a generic PAKE framework based on smooth projective hash functions (SPHF) which was introduced by Cramer and Shoup [17] to construct a CCA-security PKE. After Gennaro and Lindell's ingenious work, a plenty of more efficient PAKE protocols has been constructed from various SPHFs [1, 2, 8, 15, 22].

For instance, a post-quantum cryptography competition is held by NIST to advance the process of post-quantum cryptography standard, and lattice-based cryptosystems is the most promising candidates among lattice-based cryptosystems, code-based cryptosystems, multivariate-based cryptosystems and hash-based cryptosystems. The learning with errors (LWE) problem, which was first proposed by Regev [33] as an extension of learning parity with noise (LPN) problem [10], is one of the most widely used lattice problem to construct lattice-based cryptography. Later in [29], Lyubashevsky et al. introduced the RLWE, and proved the hardness of RLWE. (R)LWE has attracted a lot of attention in theories and applications due to its good asymptotical efficiency, strong security and exquisite construction. (R)LWE has been used to construct public-key encryption [20, 28, 33], identity-based encryption [3, 16], (authenticated) key exchange [4, 11, 19, 35], and fully homomorphic encryption [13, 14], etc.

However, it was still an open problem to construct a lattice-based PAKE from SPHF until Katz and Vaikuntanathan [25] gave the first lattice-based PAKE from a weaker notion of SPHF— ϵ -approximate SPHF (ASPHF). Later, Zhang and Yu [34] construct a two-round lattice-based PAKE from splittable PKE with associated ASPHF. In [27], Li and Wang constructed a two-round lattice-based PAKE inspired by [1]. A drawback shared by all of these PAKEs is that they are all asymmetric, which means that the session key is completely decided by only one party. The common structure of these lattice-based PAKEs can be summarized as follow: both client and server compute a string in \mathbb{Z}_2^n , assume tk and tk' , whose hamming distance are not too far. Then, server (client) chooses a session key from $\{0, 1\}^\kappa$, and sends $\Delta = tk \oplus ECC(sk)$ to client (server). Finally, client (server) computes $sk = ECC^{-1}(tk \oplus \Delta)$.

In this paper, we adopt the technique of reconciliation mechanisms instead of the ECC. Reconciliation mechanism is firstly introduced by Ding et al. [19] to

construct the first lattice-based KE from (R)LWE. The different between reconciliation mechanisms and ECC is that the input of reconciliation mechanisms is in \mathbb{Z}_q , and \mathbb{Z}_{2^n} for ECC. The benefits of using reconciliation mechanisms is: (1) the new PAKE is a symmetric protocol (the session key contain the information of both participants instead of just one party). (2) suppose the input of ECC is k bits, the output is n bits, and the code is capable of correcting t bits errors. The message (Δ in the original PAKEs and ω in ours) to compute a consistent session key for two party is at least $2t + 1$ bits shorter than ECC. In the original applications, $t = 2\epsilon$, where ϵ is the parameter of ASHPF. (3) the dimensions of hash keys, project keys and the output of ASHPF is k , the dimension is at least $2t + 1$ shorter than the situation when using ECC, which means we need less computation to achieving the same security level.

The rest of this paper is organized as follows. In Sect. 2, we introduce some notations and reconciliation mechanisms, and also introduce the framework of the two-round PAKE from [34] In Sect. 3, we introduce our new symmetric PAKE in details, and show the correctness and security of the new PAKE. Finally, Sect. 4 concludes this paper.

2 Preliminaries

Let κ be the security parameter. Bold capital letters denote matrices. Bold lowercase letters denote vectors. The length of a vector is denoted by $\|\cdot\|$. For any integer q , let \mathbb{Z}_q denote the quotient ring $\mathbb{Z}/q\mathbb{Z}$. We use $a \leftarrow_r B$ to denote that a is an element randomly chosen from B , where B is a distribution or a finite set. When we say that a function $f(x)$ is negligible, we mean that for every $c > 0$, there exists a X satisfies: $f(x) < 1/x^c$ for all $x > X$. The statistical distance between two distributions, X and Y , over some finite set S is defined as:

$$\Delta(X, Y) = \frac{1}{2} \sum_{s \in S} |Pr(X = s) - Pr(Y = s)|.$$

If $\Delta(X, Y)$ is negligible, we say that X and Y are statistically indistinguishable.

2.1 Lattice and Gaussian Distributions

A lattice always connects to a matrix \mathbf{B} , and it is finitely generated as the integer linear combinations of the column vectors of $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_k\}$:

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^n z_i \mathbf{b}_i : z_i \in \mathbb{Z} \right\}.$$

The integer n is called the rank of the basis, and it is an invariant of the lattice. For any positive integer n and real $s > 0$, define the Gaussian function of parameter s as:

$$\rho_s(\mathbf{x}) = \exp(-\pi \|\mathbf{x}\|/s^2).$$

We define a Gaussian distribution over lattice \mathcal{L} as:

$$D_s(\mathbf{x}) = \rho_s(\mathbf{x}) / \rho_s(\mathcal{L}).$$

where $\rho_s(\mathcal{L}) = \sum_{\mathbf{x} \in \mathcal{L}} \rho_s(\mathbf{x})$.

Lemma 1. [31] For any positive integer $m \in \mathbb{Z}$, and large enough $s \geq \omega(\sqrt{\log m})$. Suppose $\mathbf{x} \leftarrow_r D_{\mathbb{Z}^m, s}$, it holds that:

$$\Pr(\|\mathbf{e}\| > s\sqrt{m}) \leq 2^{-m+1}.$$

This fact shows that the probability that the samples from Gaussian distribution are not around the mean is small.

2.2 Reconciliation Mechanism

Reconciliation mechanism was first proposed by Ding et al. [19] and later be reconstructed by a series of works [4, 23, 32]. It enables two parties to extract identical information from two almost same elements σ_1 and $\sigma_2 \in \mathbb{Z}_q$. In our protocol, the reconciliation mechanism OKCN [23] is adopted, and a brief description of OKCN is given as follows.

The OKCN consists of two algorithms (*Con*, *Rec*) which have parameters q (dominating security and efficiency), m (parameterizing range of consensus key), g (parameterizing bandwidth), and d (parameterizing error rate). Define $params = (q, m, g, d, aux)$ where $aux = (q' = lcm(q, m), \alpha = q'/q, \beta = q'/m)$. The probabilistic polynomial time algorithm *Con* takes a security parameter $(\sigma_1, params = (q, m, g, d))$ as input and outputs (k_1, ω) where $k_1 \in \mathbb{Z}_m$ is the shared value and $\omega \in \mathbb{Z}_g$ is the signal that will be publicly delivered to the communicating peer. The deterministic algorithm *Rec*, on input $(\sigma_2, \omega, params)$, outputs k_2 which is identical to k_1 with overwhelming probability. The details of OKCN are presented in Algorithm 1.

Algorithm 1. Reconciliation Mechanism: OKCN

```

1: function CON( $\sigma_1, params$ )
2:    $e \leftarrow [-\lfloor(\alpha - 1)/2\rfloor, \lfloor\alpha/2\rfloor]$ 
3:    $\sigma_A = (\alpha\sigma_1 + e) \bmod q'$ 
4:    $k_1 = \lfloor\sigma_A/\beta\rfloor$ 
5:    $\omega = \lfloor(\sigma_A \bmod \beta)g/\beta\rfloor \in \mathbb{Z}_g$ 
6:   return  $(k_1, \omega)$ 
7: end function
8: function REC( $\sigma_2, \omega, params$ )
9:    $k_2 = \lfloor\alpha\sigma_2/\beta - (\omega + 1/2)/g\rfloor \bmod m$ 
10:  return  $k_2$ 
11: end function

```

Lemma 2. [23] For OKCN: (1) k_1 and ω are independent, and k_1 is uniformly distributed over \mathbb{Z}_m , whenever $\sigma_1 \leftarrow \mathbb{Z}_q$; (2) If the system parameters satisfy $(2d + 1)m < q(1 - 1/g)$ where $m \geq 2$ and $g \geq 2$, then the OKCN is correct ($k_1 = k_2$).

2.3 Approximate Smooth Projective Hash Functions

Smooth projective hash functions (SPHF) were first defined by Cramer and Shoup [17] for achieving CCA-secure PKEs. Approximate smooth projective hash functions (ASPHF) was defined by Katz and Vaikuntanathan [25] to match the requirement of lattice hard assumptions. Roughly speaking, the main difference between SPHF and ASPHF is that ASPHF require only approximate correctness.

Formally, let $PKE = (KeyGen, Enc, Dec)$ be a CCA-security PKE scheme, and let \mathcal{P} be an efficiently recognizable plaintext space of PKE. Then we define three sets:

$$\begin{aligned} X &= \{(label, c, pw) \mid (label, c) \in C_{pk}; pw \in \mathcal{P}\}; \\ L &= \{(label, c, pw) \in X \mid label \in \{0, 1\}^*\}; \\ &\quad c = Enc(pk, label, pw)\}; \\ \bar{L} &= \{(label, c, pw) \in X \mid label \in \{0, 1\}^*\}; \\ &\quad pw = Dec(sk, label, c)\}. \end{aligned}$$

Definition 1. (ϵ -approximate SPHF). An ϵ -approximate SPHF is defined by a sampling algorithm that, given a public key pk of $\mathcal{PK}\mathcal{E}$, outputs $(K, \{H_{hk} : X \rightarrow \mathbb{Z}_q^n\}_{hk \in K}, S, Proj : K \rightarrow S)$. It contains three efficient algorithms: (1) sampling a hash key $hk \leftarrow K$. (2) computing $H_{hk}(x) = H_{hk}(c, pw)$ for all $hk \in K$ and $x = (label, c, pw) \in X$. (3) computing $hp = Proj(hk)$ for all $hk \in K$.

- (approximate correctness) For all $x = (label, c, pw) \in L$ and randomness r , there exists an efficient algorithm computing the value $Hash(hp, x, r) = Hash(hp, c, pw, r)$, and satisfies $Pr[|H_{hk}(c, pw) - Hash(hp, c, pw, r)| \geq \epsilon \cdot q] = \text{negl}(\kappa)$ over the choice of $hk \in K$.
- (smoothness) For any function $h : S \rightarrow X \setminus \bar{L}$, $hk \leftarrow K$, $hp = Proj(hk)$, $x = h(hp)$ and $\rho \leftarrow \mathbb{Z}_q^n$, the statistical distance between $(hp, H_{hk}(x))$ and (hp, ρ) is negligible in the security parameter κ .

Compared to the ASPHF notion in [34], our definition of ASPHF mainly has two modifications. Firstly, the output of H_{hk} and $Hash(hp, c, pw, r)$ are in \mathbb{Z}_q^r instead of $\{0, 1\}^\ell$. Secondly, the approximate parameter ϵ depend on the distance of two element in \mathbb{Z}_q instead of the hamming distance of two element in $\{0, 1\}^\ell$.

2.4 Two-Round PAKE in [34]

At Asiacrypt 2017, Zhang and Yu proposed a two-round PAKE from splittable PKEs with associated ASPHF inspired by the works of Katz and Vaikuntanathan

[25, 26], and they instantiated the splittable PKEs with associated ASPHF based on the LWE assumption. Now, we show the execution of their protocol in details in Fig. 1.

Let (Gen, Enc, Dec) be a CCA-secure labeled public-key encryption scheme, we have $c = (u, v) = Enc(pk, label, pw, r)$, where $u = f(pk, pw, r)$ and $v = g(pk, label, pw, r)$. Let $(K, \ell, \{H_k : X \rightarrow \{0, 1\}^\ell\}_{k \in K}, S, Proj : K \rightarrow S)$ be an associated ϵ -approximate smooth projective hash function. We adopt an error-correcting code to cope with the tiny errors between two parties. Let $ECC : \{0, 1\}^\kappa \rightarrow \{0, 1\}^\ell$ be an error-correcting code which can correct 2ϵ fraction of errors. The $ECC^{-1} : \{0, 1\}^\ell \rightarrow \{0, 1\}^\kappa$ is the decoding algorithm. We assume that for uniformly distributed $\rho \in \{0, 1\}^\ell$, the distribution of $\omega = ECC^{-1}(\rho)$ is uniform over $\{0, 1\}^\kappa$.

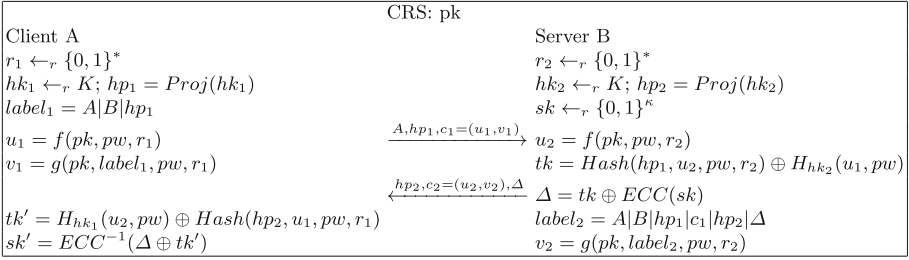


Fig. 1. 2-round PAKE from ASHPF in [34]

3 Symmetric PAKE

3.1 Modified Splittable PKE with Associated ASPHF

First of all, we slightly modify the splittable PKEs with associated ASPHF to accommodate the reconciliation mechanism. Let $(CRSGen; Prove; Verify)$ be a simulation-sound NIZK proof for R_{pke} which proves that c_1 and c_2 encrypt the same w . The $\mathcal{PK}\mathcal{E} = (KeyGen, Enc, Dec)$ from [34] is defined as follows.

- $KeyGen(1^\kappa)$: Compute $(\mathbf{A}_0, \mathbf{R}_0) \leftarrow TrapGen(1^n; 1^m, q)$ and $(\mathbf{A}^1, \mathbf{R}_1) \leftarrow TrapGen(1^n, 1^m, q)$ and $crs \leftarrow CRSGen(1^\kappa)$. Return $(pk, sk) = ((\mathbf{A}_0, \mathbf{A}_1, crs), \mathbf{R}_0)$.
- $Enc(pk, label, w)$: Randomly choose $s_0, s_1 \leftarrow_r \mathbb{Z}_q^{n_1}$, $e_0, e_1 \leftarrow_r D_{\mathbb{Z}^m, \alpha q}$. Finally, return the ciphertext $C = (c_0, c_1, \pi)$, where

$$c_0 = \mathbf{A}_0^t \begin{pmatrix} s_0 \\ \mathbf{1} \\ w \end{pmatrix} + e_0, \quad c_1 = \mathbf{A}_1^t \begin{pmatrix} s_0 \\ \mathbf{1} \\ w \end{pmatrix} + e_1,$$

and $\pi = Prove(crs, (\mathbf{A}_0, \mathbf{A}_1, c_0, c_1, \beta), (s_0, s_1, w), label)$.

– $Dec(sk, label, C)$:

If $Verify(crs; (A_0, A_1, c_0, c_1, \beta), \pi, label) = 0$, return \perp . Otherwise, compute

$$\mathbf{t} = \begin{pmatrix} s_0 \\ 1 \\ \mathbf{w} \end{pmatrix} \leftarrow Solve(\mathbf{A}_0, \mathbf{R}_0, \mathbf{c}_0),$$

and return $\mathbf{w} \in \mathbb{Z}_q^n$.

Then, we construct a new associated ASHPF $(K, \ell, \{H_{hk} : X \rightarrow \{0, 1\}^\ell\}_{hk \in K}, S, Proj : K \rightarrow S)$ for the above $\mathcal{PK}\mathcal{E}$ as follows:

- The hash key $hk = (\mathbf{x}_1, \dots, \mathbf{x}_\ell)$, where $\mathbf{x}_i \leftarrow D_{\mathbb{Z}^m}$, and $\mathbf{A}_0^t = (\mathbf{B} || \mathbf{U}) \in \mathbb{Z}_q^{m \times n}$ where $\mathbf{B} \in \mathbb{Z}_q^{m \times n_1}$ and $\mathbf{U} \in \mathbb{Z}_q^{m \times (n_2+1)}$. The projection key $hp = Proj(hk) = (\mathbf{u}_1, \dots, \mathbf{u}_\ell)$, where $\mathbf{u}_i = \mathbf{B}^t \mathbf{x}_i$.
- $H_{hk}(x) = H_{hk}((c_0; c_1); w)$: Given $hk = (\mathbf{x}_1, \dots, \mathbf{x}_\ell)$ and $x = (label, C, \mathbf{w}) \in X$ for some $C = (\mathbf{c}_0, \mathbf{c}_1, \pi)$, compute

$$z_i = \mathbf{x}_i^t \left(\mathbf{c}_0 - \mathbf{U} \begin{pmatrix} 1 \\ \mathbf{w} \end{pmatrix} \right) \in \mathbb{Z}_q$$

for $i \in \{1, \dots, \ell\}$. Finally, return $H_{hk}((\mathbf{c}_0, \mathbf{c}_1), \mathbf{w}) = (z_1, \dots, z_\ell)$.

- $Hash(hp, ((\mathbf{c}_0, \mathbf{c}_1), \mathbf{w}), \mathbf{s}_0)$: Given $hp = (\mathbf{u}_1, \dots, \mathbf{u}_\ell)$, $x = (label, (\mathbf{c}_0, \mathbf{c}_1, \pi), \mathbf{w})$ and $\mathbf{s}_0 \in \mathbb{Z}_q^{n_1}$, compute $z_i = \mathbf{u}_i^t \mathbf{s}_0$. Finally, return $Hash(hp, ((\mathbf{c}_0, \mathbf{c}_1), \mathbf{w}), \mathbf{s}_0) = (z_1, \dots, z_\ell)$.

Firstly, it is obviously that these algorithms are efficient, and the smoothness of the ASHPF directly follows the Theorem 3 of [34]. Then, we consider the correctness of the ASHPF:

$$\begin{aligned} |z_i - z'_i| &= \left| \mathbf{x}_i^t \left(\mathbf{c}_0 - \mathbf{U} \begin{pmatrix} 1 \\ \mathbf{w} \end{pmatrix} \right) - \mathbf{u}_i^t \mathbf{s}_0 \right| \\ &\leq |\mathbf{x}_i^t \mathbf{e}_0| \\ &\leq m\gamma\alpha q \end{aligned}$$

The second inequality is from the Lemma 1. So the above ASHPF is correct, and $\epsilon = m\gamma\alpha$. Finally, we conclude the property of our ASHPF as the following theorem.

Theorem 1. *Let $\epsilon = m\gamma\alpha$, and let $n, m, q, \alpha, \beta, \gamma$ be defined as above. Let ℓ be polynomial in the security parameter κ . Then, $(K, \ell, \{H_{hk} : X \rightarrow \{0, 1\}^\ell\}_{hk \in K}, S, Proj : K \times C_{pk} \rightarrow S)$ is an ϵ -approximate ASPH.*

3.2 Framework of Symmetric PAKE

The PAKE protocol we construct works in the CRS model. The protocol was initiated by a client A , and after the protocol, both client A and server B holding

	CRS: pk	
Client A		Server B
$r_1 \leftarrow_r \{0, 1\}^*$		$r_2 \leftarrow_r \{0, 1\}^*$
$hk_1 \leftarrow_r K; hp_1 = Proj(hk_1)$		$hk_2 \leftarrow_r K; hp_2 = Proj(hk_2)$
$label_1 = A B hp_1$		
$u_1 = f(pk, pw, r_1)$	$\xrightarrow{A, hp_1, c_1 = (u_1, v_1)}$	$u_2 = f(pk, pw, r_2)$
$v_1 = g(pk, label_1, pw, r_1)$		$tk = Hash(hp_1, u_2, pw, r_2) + H_{hk_2}(u_1, pw)$
	$\xleftarrow{hp_2, c_2 = (u_2, v_2), \omega}$	$(sk, \omega) \leftarrow Con(tk, params)$
$tk' = H_{hk_1}(u_2, pw) + Hash(hp_2, u_1, pw, r_1)$		$label_2 = A B hp_1 c_1 hp_2 \omega$
$sk' = Rec(tk', \omega, params)$		$v_2 = g(pk, label_2, pw, r_2)$

Fig. 2. Framework of symmetric PAKE

the same pw compute an consistent session key. A high-level overview of the protocol is given in Fig. 2, and a detailed description follows.

Setup. The CRS of our symmetric PAKE protocol consists of the public key pk of the encryption scheme we used, which can be generated by a trusted third party. No party need to know the secret key sk corresponding to pk as there is no decryption in our protocol.

Protocol Execution. Consider an execution of the protocol between a client A and a server B . Suppose \mathcal{D} is the set of valid passwords. A and B hold a shared password $pw \in \mathcal{D}$.

- **First round.** The client A chooses a hash key $hk_1 \leftarrow_r K$ for the ASPHF, and computes the projection key $hp_1 = Proj(hk_1)$. Then C defines $label_1 = A|B|hp_1$, and computes $u_1 = f(pk, pw, r_1)$ and $v_1 = g(pk, label_1, pw, r_1)$ where $r_1 \leftarrow \{0, 1\}^*$ is the random coins of Enc . Finally, A sends $(A, hp_1, c_1 = (u_1, v_1))$ to the server B .
- **Second round.** Upon receiving $(A, hp_1, c_1 = (u_1, v_1))$ from the client A , the server B checks if c_1 is a valid ciphertext with respect to pk and $label_1$. If not, B rejects and aborts. Otherwise, B chooses a hash key $hk_2 \leftarrow_r K$, and computes the projection key $hp_2 = Proj(hk_2)$. Then B computes $u_2 = f(pk, pw, r_2)$ where $r' \leftarrow \{0, 1\}^*$. After that, B computes $tk = Hash(hp_1, u_2, pw, r_2) + H_{hk_2}(u_1, pw)$ ¹, and $(sk, \omega) \leftarrow Con(tk, params)$. B defines $label_2 = A|B|hp_1|hp_2|\omega$, and computes $v_2 = g(pk, label_2, pw, r_2)$. Finally, B sends $(hp_2, c_2 = (u_2, v_2), \omega)$ to the server A .
- **Local computation.** After receiving $(hp_2, c_2 = (u_2, v_2), \omega)$ from server B , the client A checks if c_2 is a valid ciphertext with respect to pk and $label_2$. If not, A rejects and aborts. Otherwise, A computes $tk' = H_{hk_1}(u_2, pw) + Hash(hp_2, u_1, pw, r_1)$. Finally, A computes $sk' = Rec(tk', \omega, params)$.

3.3 Analysis of the Protocol

In the following, we say that a user (or an instance of a user) accepts an incoming message msg as a valid protocol message if no abort happens. A client/server will obtain a session key only if he receives a valid protocol message.

¹ The additions in this paper are performed in \mathbb{Z}_q .

Theorem 2. *If (Gen, Enc, Dec) is a CCA-secure labeled public-key encryption scheme with an associated ϵ -approximate smooth projective hash function, and $OKCN : (Con, Rec)$ is a reconciliation mechanism which can correct $d \geq 2\epsilon \cdot q$ of errors. Then, after an run of the above PAKE protocol, honestly users can obtain the same session key with overwhelming probability.*

Proof. First of all, the ciphertexts generated in an honest execution of the protocol are valid. Now we show that client and server compute the identical session keys with overwhelming probability.

$$\begin{aligned} tk - tk' &= \left(Hash(hp_1, u_2, pw, r_2) + H_{hk_2}(u_1, pw) \right) \\ &\quad - \left(H_{hk_1}(u_2, pw) + Hash(hp_2, u_1, pw, r_1) \right) \\ &= \left(Hash(hp_1, u_2, pw, r_2) - H_{hk_1}(u_2, pw) \right) \\ &\quad + \left(H_{hk_2}(u_1, pw) - Hash(hp_2, u_1, pw, r_1) \right) \end{aligned}$$

ϵ -approximate correctness of the ASPHF implies that $|Hash(hp_1, u_2, pw, r_2) - H_{hk_1}(u_2, pw)| \leq \epsilon \cdot q$ with overwhelming probability. Similarly, $|H_{hk_2}(u_1, pw) - Hash(hp_2, u_1, pw, r_1)| \leq \epsilon \cdot q$. Thus, we have $|tk - tk'| \leq 2\epsilon \cdot q$. With overwhelming probability, we have

$$sk' = Rec(tk', \omega, params) = sk.$$

as the *OKCN* we adopt can correct at least $2\epsilon \cdot q$ errors.

The security proof strategy of our protocol is similar to that in [34] which shows the security of a two-round PAKE protocol from ASHPF and ECC, but they are inherently different. The security of the protocol we construct is based on the first property of *OKCN* in Lemma 2 instead of ECC in [34]. We conclude the security of our protocol with the following theorem.

Theorem 3. *If (Gen, Enc, Dec) is a CCA-secure labeled public-key encryption scheme with an associated ϵ -approximate smooth projective hash function, $OKCN : (Con, Rec)$ is a reconciliation mechanism which can correct $d \geq 2\epsilon \cdot q$ of errors. Then, the protocol in Fig. 2 is a security PAKE protocol.*

Proof. Here we just give some intuitions of the proof. First we note that for a passive adversary (or a adversary try to obtain any useful information of the real password pw via the Execute query), the shared session-key is pseudorandom. As in [26, 34], we assume 0 is not a valid password. By the semantic security of the PKE scheme, it is computationally indistinguishable for the adversary if the ciphertext is a encryption of the real pw or 0. By the smoothness of the ASPHF, the shared session keys are pseudorandom as $0 \notin \mathcal{D}$.

Now we consider an active adversary. First of all, the adversary can not construct a new valid ciphertext that decrypts to the real password pw with probability more than $q/|\mathcal{D}|$ by the CCA-security of PKE. Otherwise, if the

adversary sends the client a ciphertext that does not decrypt to the real password pw , then the session keys computed are indistinguishable from uniform in the adversary's view by the smoothness of the ASPHF. Thus, for $Q(k)$ times on-line attacks, the advantage of the adversary is at most $Q(k)/|D|$.

We would like to emphasize that the techniques we use here to construct a symmetric PAKE can be applied to other lattice-based PAKEs [25, 27]. They can also reduce the size of hash keys and the communication complexity, and simplify these PAKEs. We omit the details here.

4 Conclusion

In this paper, we improve the lattice-based PAKE in [34] with technique of reconciliation mechanism, and we show that the computation complexity and communication complexity of the new PAKE are reduced. We emphasize that this technique is also useful to all other lattice-based PAKEs. To accommodate the reconciliation mechanism to our application, we give a modified definition of ASPHF. Moreover, with the help of reconciliation mechanism, our PAKEs are symmetric. We show the correctness and security of our PAKEs.

References

1. Abdalla, M., Benhamouda, F., Pointcheval, D.: Public-key encryption indistinguishable under plaintext-checkable attacks. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 332–352. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_15
2. Abdalla, M., Chevalier, C., Pointcheval, D.: Smooth projective hashing for conditionally extractable commitments. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 671–689. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_39
3. 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
4. Alkim, E., Ducas, L., Poppelmann, T., Schwabe, P.: Post-quantum key exchange—a new hope. In: USENIX Security Symposium, pp. 327–343 (2016)
5. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_11
6. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48329-2_21
7. Bellare, M., Merritt, M.: Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. IEEE Computer Society (1992)
8. Benhamouda, F., Blazy, O., Chevalier, C., Pointcheval, D., Vergnaud, D.: New techniques for SPHF and efficient one-round PAKE protocols. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 449–475. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_25

9. Boneau, J.: The science of guessing: analyzing an anonymized corpus of 70 million passwords. In: Security & Privacy. IEEE (2012)
10. Berlekamp, E.R., McEliece, R.J., Van Tilborg, H.C.: On the inherent intractability of certain coding problems. *IEEE Trans. Inf. Theory* **24**(3), 384–386 (1978)
11. Bos, J., Costello, C., Naehrig, M., Stebila, D.: Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In: IEEE Symposium on Security and Privacy, pp. 553–570 (2015)
12. Boyko, V., MacKenzie, P., Patel, S.: Provably Secure password-authenticated key exchange using Diffie-Hellman. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 156–171. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_12
13. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. *SIAM J. Comput.* **43**(2), 831–871 (2014)
14. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from Ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_29
15. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.: Universally composable password-based key exchange. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 404–421. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_24
16. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. *J. Cryptol.* **25**(4), 601–639 (2012)
17. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_4
18. Diffie, W., Hellman, M.: New directions in cryptography. *IEEE Trans. Inf. Theory* **22**(6), 644–654 (1976)
19. Ding, J., Xie, X., Lin, X.: A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem, <http://eprint.iacr.org/2012/688> (2012)
20. 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 (STOC 2008), pp. 197–206. ACM, New York (2008)
21. Gennaro, R., Lindell, Y.: A framework for password-based authenticated key exchange. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_33
22. Groce, A., Katz, J.: A new framework for efficient password-based authenticated key exchange. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, 4–8 October 2010
23. Jin, Z., Zhao, Y.: Optimal Key Consensus in Presence of Noise. <http://eprint.iacr.org/2017/1058> (2017)
24. Katz, J., Ostrovsky, R., Yung, M.: Efficient password-authenticated key exchange using human-memorable passwords. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 475–494. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44987-6_29
25. Katz, J., Vaikuntanathan, V.: Smooth projective hashing and password-based authenticated key exchange from lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 636–652. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_37

26. Katz, J., Vaikuntanathan, V.: Round-optimal password-based authenticated key exchange. *J. Cryptol.* **26**(4), 714–743 (2013)
27. Li, Z., Wang, D.: Two-round PAKE protocol over lattices without NIZK. In: Guo, F., Huang, X., Yung, M. (eds.) *Inscrypt 2018*. LNCS, vol. 11449, pp. 138–159. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-14234-6_8
28. Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-based encryption. In: Kiayias, A. (ed.) *CT-RSA 2011*. LNCS, vol. 6558, pp. 319–339. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19074-2_21
29. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. *J. ACM* **60**(6), 1–35 (2013)
30. MacKenzie, P., Patel, S., Swaminathan, R.: Password-authenticated key exchange based on RSA. In: Okamoto, T. (ed.) *ASIACRYPT 2000*. LNCS, vol. 1976, pp. 599–613. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44448-3_46
31. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. In: *Annual Symposium on Foundations Of Computer Science*. IEEE Computer Society (2004)
32. Peikert, C.: Lattice cryptography for the internet. In: Mosca, M. (ed.) *PQCrypto 2014*. LNCS, vol. 8772, pp. 197–219. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11659-4_12
33. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. *J. ACM* **56**(6), 506–519 (2009)
34. Zhang, J., Yu, Y.: Two-round PAKE from approximate SPH and instantiations from lattices. In: Takagi, T., Peyrin, T. (eds.) *ASIACRYPT 2017*. LNCS, vol. 10626, pp. 37–67. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70700-6_2
35. Zhang, J., Zhang, Z., Ding, J., Snook, M., Dagdelen, Ö.: Authenticated key exchange from ideal lattices. In: Oswald, E., Fischlin, M. (eds.) *EUROCRYPT 2015*. LNCS, vol. 9057, pp. 719–751. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_24



Zero-Knowledge Proofs for Improved Lattice-Based Group Signature Scheme with Verifier-Local Revocation

Yanhua Zhang¹(✉), Yifeng Yin¹, Ximeng Liu², Qikun Zhang¹, and Huiwen Jia³

¹ Zhengzhou University of Light Industry, Zhengzhou 450002, China
{yhzhang,yinyifeng,kzhang}@zzuli.edu.cn

² Fuzhou University, Fuzhou 350108, China
snbnix@gmail.com

³ Guangzhou University, Guangzhou 510006, China
hwjia@gzhu.edu.cn

Abstract. The first lattice-based group signature scheme with verifier-local revocation (GS-VLR) was introduced by Langlois et al. in PKC 2014, and subsequently, a full and corrected version was designed by Ling et al. in TCS 2018. However, zero-knowledge proofs in both schemes are within a structure of *Bonsai Tree*, and have bit-sizes of the group public-key and the member secret-key proportional to $\log N$, where N is the group size. On the other hand, the revocation tokens in both schemes are related to some public matrix and the group member secret-key, and thus only obtain a weaker security, *selfless-anonymity*. For the tracing algorithms in both schemes, they just run in the linear time of N . Therefore, for a large group, zero-knowledge proofs in lattice-based GS-VLR schemes are not that secure and efficient.

In this work, we firstly utilize an efficient and compact *identity-encoding technique* which only needs a constant number of public matrices to encode the member's identity information and it saves a $\mathcal{O}(\log N)$ factor in both bit-sizes for the group public-key and the group member secret-key. Secondly, separating from the member secret-key, we generate revocation token within some secret Gaussian vector and thus obtain a stronger security, *almost-full anonymity*. Moreover, the explicit traceability, to trace the signer's identity in a constant time, independent of N , for the tracing authority is also satisfied. In particular, a new Stern-type statistical zero-knowledge proofs protocol for an improved lattice-based GS-VLR scheme enjoying the above three advantages is proposed.

Keywords: Lattice-based group signatures · Verifier-local revocation · Zero-knowledge proofs · Explicit traceability · Almost-full anonymity

1 Introduction

Group signature (GS), first introduced by Chaum and van Heyst [8], is a fundamental privacy-preserving primitive that allows a group member to issue signatures on behalf of the whole group without compromising its identity information

(*anonymity*); given a valid message-signature pair (M, Σ) , a tracing authority can reveal the signer’s real identity (*traceability*). Thus, these two excellent properties allow GS to find several real-life applications, such as in trusted computing, anonymous road-to-vehicle online communications, digital right managements, e-commerce systems, and much more.

Generally, to construct an efficient group signature three critical cryptographic ingredients are required: a digital signature (DS) scheme, a public-key encryption (PKE) scheme, a zero-knowledge proofs (ZKP) protocol. Therefore to design a theoretical secure and efficient GS scheme is a challenging work for the research community and over the last quarter-century GS schemes with different security notions, different levels of efficiency and based on different hardness assumptions have been proposed (e.g., [2–5, 11, 13] ...).

LATTICE-BASED GS-VLR. Lattice-based cryptography (LBC), believed to be the most promising candidate for post-quantum cryptography, enjoys several noticeable advantages over conventional number-theoretic cryptography (e.g., based on integer factoring or discrete logarithm problems): conjectured resistance against quantum computers, faster arithmetic operations and security under the *worst-case* hardness assumptions. Since the creative works of Ajtai [1], Regev [31] and Gentry et al. [10], LBC has attracted significant interest by the research community and become an exciting cryptographic research field. In recent ten years, along with other lattice-based primitives, GS has been paid a greet attention and since the first construction introduced by Gordon et al. [11], a series of lattice-based GS schemes with static or dynamic design techniques [6, 14, 16–18, 21–24, 26] were proposed.

As an orthogonal problem of member enrollment (ME), the support for membership revocation (MR) is another desirable functionality for lattice-based GS. The verifier-local revocation (VLR) mechanism, which only requires the verifiers to possess up-to-date group information (i.e., a revocation list, RL, consists of a series of revocation tokens for revoked members), but not the signers, is much more efficient than accumulators, especially when considering a large group, and the first lattice-based construction was introduced by Langlois et al. [15] in PKC 2014, and subsequently, a full and corrected version was proposed by Ling et al. [19] in TCS 2018, furthermore, two schemes achieving different security notions (almost-full anonymity *v.s.* dynamical-almost-full anonymity) were constructed by Perera and Koshiba [28, 29].

However, all mentioned lattice-based GS-VLR schemes are within the structure of *Bonsai Tree* [7], and thus features bit-sizes of the group public-key and the group member secret-key proportional to $\log N$, where N is the group size, the maximum number of group members. The only two exceptions are [9, 32] which adopt an identity-encoding function introduced in [26] to encode the member’s identity index and save a $\mathcal{O}(\log N)$ factor for both bit-sizes. However, the latter two constructions involve a series of sophisticated encryptions and ZKP protocols in the signing phase, on the other hand, revocation tokens in [9, 15, 19, 32] are all related to some public matrix and the group member secret-key (the product of a public matrix and one part of the member secret-key), thus all schemes only

obtain a weaker security, *selfless-anonymity*, as introduced in [4]. For the tracing algorithms in [9, 15, 19, 32], they all just run in a linear time in N (i.e., one by one for group members until the signer is traced). Therefore for a large group, ZKP protocols in lattice-based GS-VLR are not that secure and efficient. These somewhat unsatisfactory state-of-affairs highlights a challenge to construct some simpler and efficient lattice-based GS-VLR schemes, in particular, to design the efficient statistical ZKP protocols corresponding to these constructions.

OUR RESULTS AND TECHNIQUES. In this work, we reply positively to the problems discussed above. Specifically, we pay attention to a new design of Stern-type statistical ZKP protocol for an improved lattice-based GS-VLR scheme. Firstly, by adopting an efficient and compact identity-encoding technique, the bit-sizes of the group public-key and member secret-key can save a $\mathcal{O}(\log N)$ factor in comparison with the existing lattice-based schemes. Secondly, separating from the member secret-key, the revocation token is generated within a secret Gaussian vector, and thus obtaining almost-full anonymity, a stronger security. Thirdly, based on a lattice-based verifiable encryption protocol corresponding to the dual learning with errors (LWE) cryptosystem, the explicit traceability (ET), used to trace the signer's identity in a constant time, independent of N , is also satisfied.

We also declare that the new and efficient Stern-type statistical ZKP protocol for an improved lattice-based GS-VLR scheme with the shorter key-sizes, stronger security and explicit traceability can be obtained in a relatively simple manner, thanks to three main techniques discussed below.

Firstly, to realize a simpler and efficient Stern-type statistical ZKP protocol for lattice-based GS-VLR with the shorter key-sizes, some efficient mechanism is required to encode the group member's identity information. We utilize a compact identity-encoding technique as in [26] which only needs a constant number of public matrices to encode the group member's identity index. We consider the group of $N = 2^\ell$ members and each group member is identified by a ℓ -bits string $\text{id} = (d_1, d_2, \dots, d_\ell) \in \{0, 1\}^\ell$ which is a binary representation of its index $i \in \{1, 2, \dots, N\}$, i.e., $\text{id} = \text{bin}(i) \in \{0, 1\}^\ell$. In our new Stern-type ZKP protocol (without the structure of *Bonsai Tree*), the group public-key only consists of one random vector $\mathbf{u} \in \mathbb{Z}_q^n$ and four random matrices $\mathbf{A}_0, \mathbf{A}_1^1, \mathbf{A}_2^2 \in \mathbb{Z}_q^{n \times m}$ (used for identity-encoding) and $\mathbf{A}_3^3 \in \mathbb{Z}_q^{n \times m}$ (only used for explicit traceability).

For member i , instead of generating a short trapdoor basis matrix for a hard random lattice as the signing secret-key as in [26], we sample a $2m$ -dimensional Gaussian vector $\mathbf{e}_i = (\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) \in \mathbb{Z}^{2m}$ satisfying $0 < \|\mathbf{e}_i\|_\infty \leq \beta$, $\mathbf{A}_i \cdot \mathbf{e}_i = \mathbf{u} \bmod q$, where $\mathbf{A}_i = [\mathbf{A}_0 | \mathbf{A}_1^1 + i\mathbf{A}_2^2] \in \mathbb{Z}_q^{n \times 2m}$. Furthermore, for the VLR feature to obtain *almost-full anonymity*, the token of i is constructed by \mathbf{A}_0 and a short Gaussian vector $\mathbf{r}_i \in \mathbb{Z}^m$ separating from $\mathbf{e}_{i,1} \in \mathbb{Z}^m$, i.e., $\text{grt}_i = \mathbf{A}_0 \cdot \mathbf{r}_i \bmod q$.

Secondly, to realize a simpler and efficient construction of Stern-type statistical ZKP protocol for lattice-based GS-VLR with explicit traceability (ET), we further need some mechanism to hide the member's index i (in our design, to hide $\text{id} = \text{bin}(i) \in \{0, 1\}^\ell$) into a ciphertext \mathbf{c} and a verifiable encryption protocol to prove that \mathbf{c} is a correct encryption of $\text{bin}(i)$.

Thus, besides the public matrices \mathbf{A}_0 , \mathbf{A}_1^1 , and \mathbf{A}_2^2 for identity-encoding, a fourth matrix \mathbf{A}_3^3 is required to encrypt $\text{bin}(i)$ using the dual LWE cryptosystem [10]. The relation then can be expressed as $\mathbf{c} = (\mathbf{c}_1 = \mathbf{A}_3^{3\top} \mathbf{s} + \mathbf{e}_1 \bmod q, \mathbf{c}_2 = \mathbf{G}^\top \mathbf{s} + \mathbf{e}_2 + \lfloor q/2 \rfloor \text{bin}(i) \bmod q)$ where \mathbf{G} is a random matrix, and \mathbf{s} , \mathbf{e}_1 , \mathbf{e}_2 are random vectors having certain specific norm.

Thirdly, the major challenge for our new Stern-type ZKP protocol lies in how to prove the following relations: (a) $[\mathbf{A}_0 | \mathbf{A}_1^1 + i \mathbf{A}_2^2] \cdot \mathbf{e}_i = \mathbf{u} \bmod q$; (b) $\text{grt}_i = \mathbf{A}_0 \cdot \mathbf{r}_i \bmod q$; (c) $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) = (\mathbf{A}_3^{3\top} \mathbf{s} + \mathbf{e}_1, \mathbf{G}^\top \mathbf{s} + \mathbf{e}_2 + \lfloor q/2 \rfloor \text{bin}(i)) \bmod q$. For relation (b), we utilize a creative idea introduced by Ling et al. [19] by drawing a matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ from the random oracle and a vector $\mathbf{e}_0 \in \mathbb{Z}^m$ from the LWE error distribution, define $\mathbf{b} = \mathbf{B}^\top \text{grt}_i + \mathbf{e}_0 = (\mathbf{B}^\top \mathbf{A}_0) \cdot \mathbf{r}_i + \mathbf{e}_0 \bmod q$, thus the member i 's token grt_i is now bound to a one-way and injective LWE function. For relation (c), we also utilize a creative idea introduced by Ling et al. [21] by constructing a matrix $\mathbf{P} \in \mathbb{Z}_q^{(m+\ell) \times (n+m+\ell)}$ (obtained from the public matrices

\mathbf{A}_3^3 and \mathbf{G} , see Sect. 3 for details), and a vector $\mathbf{e} = \begin{pmatrix} \mathbf{s} \\ \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \in \mathbb{Z}^{n+m+\ell}$, define $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) = \mathbf{P} \mathbf{e} + (\mathbf{0}^m, \lfloor q/2 \rfloor \text{bin}(i)) \bmod q$, thus the index i is bound to this new form which is convenient to construct a Stern-type statistical ZKP protocol.

For relation (a), since $\mathbf{e}_i \in \mathbb{Z}^{2m}$ is a valid solution to inhomogeneous short integer solution (ISIS) instance $(\mathbf{A}_i, \mathbf{u})$, where $\mathbf{A}_i = [\mathbf{A}_0 | \mathbf{A}_1^1 + i \mathbf{A}_2^2] \in \mathbb{Z}_q^{n \times (\ell+2)m}$, a direct way for member i to prove its validity as a certified group member without leaking \mathbf{e}_i is to perform a Stern-type statistical zero-knowledge argument of knowledge (ZKAoK) as in [20]. However, in order to protect the anonymity of i , the structure of matrix \mathbf{A}_i should not be given explicitly, thus how to realize a Stern-type statistical ZKP protocol without leaking \mathbf{A}_i and \mathbf{e}_i simultaneously? To solve this open problem, we first transform \mathbf{A}_i to \mathbf{A}' which enjoys some new form, independent of the index i , i.e., $\mathbf{A}' = [\mathbf{A}_0 | \mathbf{A}_1^1 \mathbf{g}_\ell \otimes \mathbf{A}_2^2] \in \mathbb{Z}_q^{n \times (\ell+2)m}$, where $\mathbf{g}_\ell = (1, 2^1, 2^2, \dots, 2^{\ell-1})$ is a power-of-two vector, and the index i can be rewritten as $i = \mathbf{g}_\ell^\top \cdot \text{bin}(i)$, notation \otimes denotes a concatenation with vectors or matrices, the detailed definition will be given later (see Sect. 3). A corresponding change to the signing secret-key of group member i , $\mathbf{e}_i = (\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) \in \mathbb{Z}^{2m}$ is now transformed to $\mathbf{e}'_i = (\mathbf{e}_{i,1}, \mathbf{e}_{i,2}, \text{bin}(i) \otimes \mathbf{e}_{i,2}) \in \mathbb{Z}^{(\ell+2)m}$. Thus, to argue the relation $\mathbf{A}_i \cdot \mathbf{e}_i = \mathbf{u} \bmod q$, we instead show that $\mathbf{A}' \cdot \mathbf{e}'_i = \mathbf{u} \bmod q$.

Taking all the above transformations ideas and the versatility of the Stern-type argument system introduced by Ling et al. [20] together, we can design an efficient Stern-type interactive ZKP protocol for the relations (a), (b) and (c). Furthermore, this interactive protocol can be repeated $\omega(\log n)$ times to reduce the soundness error to a negligible value, and then transformed to a secure and efficient non-interactive Stern-type statistical ZKP protocol by using the Fiat-Shamir heuristic in the random oracle model (ROM).

To summarize, by incorporating an efficient and compact identity-encoding technique, a shorter random Gaussian vector separating from the member secret-key and the lattice-based dual LWE cryptosystem to hide the identity index, a

new and efficient Stern-type statistical ZKP protocol for an improved lattice-based GS-VLR scheme is proposed, therefore, obtaining shorter key-sizes for the group public-key and the group member secret-key, almost-full anonymity, which is stronger than selfless-anonymity, and supporting the explicit traceability.

ORGANIZATION. In the forthcoming sections, we first recall some background on LBC and identity-encoding technique in Sect. 2. Section 3 turns to develop an improved identity-encoding technique, a new creation of revocation token and an explicit traceability mechanism. Our new Stern-type statistical ZKP protocol for improved lattice-based GS-VLR scheme is designed in Sect. 4, and analyzed in Sect. 5.

2 Preliminaries

2.1 Notations

\mathcal{S}_k denotes the set of all permutations of k elements, $\stackrel{\$}{\leftarrow}$ denotes that sampling elements from some given distribution uniformly at random. Let $\|\cdot\|$ and $\|\cdot\|_\infty$ denote the Euclidean norm (ℓ_2) and infinity norm (ℓ_∞) of a vector, respectively. The notation $\log a$ denotes the logarithm of a with base 2, and PPT stands for “probabilistic polynomial-time”.

2.2 Background on Lattices

For $n, m, q \geq 2$, $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{u} \in \mathbb{Z}_q^n$, the m -dimensional q -ary lattice $\Lambda_q^\perp(\mathbf{A})$ and its corresponding shift (i.e., a coset of $\Lambda_q^\perp(\mathbf{A})$) are defined as:

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{A} \cdot \mathbf{e} = \mathbf{0} \bmod q\}, \quad \Lambda_q^\mathbf{u}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{A} \cdot \mathbf{e} = \mathbf{u} \bmod q\}.$$

For real $s > 0$, define the Gaussian function on \mathbb{R}^m with a center \mathbf{c} as:

$$\forall \mathbf{e} \in \mathbb{R}^m, \quad \rho_{s,\mathbf{c}}(\mathbf{e}) = \exp(-\pi \|\mathbf{e} - \mathbf{c}\|^2 / s^2).$$

For $\mathbf{c} \in \mathbb{R}^m$, define the discrete Gaussian distribution over Λ as:

$$\forall \mathbf{e} \in \mathbb{Z}^m, \quad \mathcal{D}_{\Lambda,s,\mathbf{c}} = \rho_{s,\mathbf{c}}(\mathbf{e}) / \rho_{s,\mathbf{c}}(\Lambda) = \rho_{s,\mathbf{c}}(\mathbf{e}) / \sum_{\mathbf{e} \in \Lambda} \rho_{s,\mathbf{c}}(\mathbf{e}).$$

For convenience, we denote $\mathcal{D}_{\Lambda,s,\mathbf{c}}$ as $\mathcal{D}_{\Lambda,s}$ if $\mathbf{c} = \mathbf{0}$.

In the following, we recall some facts about discrete Gaussian distribution.

Lemma 1 ([10,30]). *For positive integers $n, q \geq 2$, and $m \geq 2n \log q$, assume that the columns of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ generate \mathbb{Z}_q^n , let $\epsilon \in (0, 1/2)$, $s \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$, then we have:*

1. For $\mathbf{e} \stackrel{\$}{\leftarrow} \mathcal{D}_{\mathbb{Z}^m,s}$, the distribution of syndrome $\mathbf{u} = \mathbf{A} \cdot \mathbf{e} \bmod q$ is within the statistical distance 2ϵ of uniform over \mathbb{Z}_q^n .
2. For $\mathbf{e} \stackrel{\$}{\leftarrow} \mathcal{D}_{\mathbb{Z}^m,s}$, $\beta = \lceil s \cdot \log m \rceil$, then $\Pr[\|\mathbf{e}\|_\infty > \beta]$ is negligible.
3. The min-entropy of $\mathcal{D}_{\mathbb{Z}^m,s}$ is at least $m - 1$.

Now we recall the definitions and hardness results for 3 *average-case* lattices problems: (inhomogeneous) short integer solution (ISIS, SIS) (in the ℓ_∞ norm) and learning with errors (LWE).

Definition 1. *The (I)SIS $_{n,m,q,\beta}^\infty$ problems are: Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a random syndrome vector $\mathbf{u} \in \mathbb{Z}_q^n$ and a real $\beta > 0$,*

- SIS $_{n,m,q,\beta}^\infty$: *to find a non-zero integer vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{e} = \mathbf{0} \pmod q$ and $\|\mathbf{e}\|_\infty \leq \beta$.*
- ISIS $_{n,m,q,\beta}^\infty$: *to find an integer vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{e} = \mathbf{u} \pmod q$ and $\|\mathbf{e}\|_\infty \leq \beta$.*

The ISIS and SIS problems are as hard as certain *worst-case* lattice problems, such as the shortest independent vectors problem (SIVP).

Lemma 2 ([10,25]). *For $m, \beta = \text{poly}(n)$, $q \geq \beta \cdot \tilde{O}(\sqrt{n})$, the average-case (I)SIS $_{n,m,q,\beta}^\infty$ are at least as hard as SIVP $_\gamma$ in the worst-case to within $\gamma = \beta \cdot \tilde{O}(\sqrt{nm})$ factor. In particular, if $\beta = 1$, $q = \tilde{O}(n)$ and $m = 2n \lceil \log q \rceil$, then the (I)SIS $_{n,m,q,1}^\infty$ problems are at least as hard as SIVP $_{\tilde{O}(n)}$.*

Definition 2. *The LWE $_{n,q,\chi}$ problem is: Given a random vector $\mathbf{s} \in \mathbb{Z}_q^n$, a probability distribution χ over \mathbb{Z} , let $\mathcal{A}_{\mathbf{s},\chi}$ be the distribution obtained by sampling a matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, a vector $\mathbf{e} \xleftarrow{\$} \chi^m$, and outputting a tuple $(\mathbf{A}, \mathbf{A}^\top \mathbf{s} + \mathbf{e})$, to distinguish $\mathcal{A}_{\mathbf{s},\chi}$ and a uniform distribution \mathcal{U} over $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$.*

Let $\beta \geq \sqrt{n} \cdot \omega(\log n)$, if q is a prime power and χ is a β -bounded distribution (e.g., $\chi^m = \mathcal{D}_{\mathbb{Z}^m, s}$), the LWE $_{n,q,\chi}$ problem is at least as hard as SIVP $_{\tilde{O}(nq/\beta)}$.

2.3 The Identity-Encoding Technique

Our new design of Stern-type statistical zero-knowledge proofs protocol builds on a compact identity-encoding technique proposed by Nguyen et al. [26]. Let's describe it briefly.

The technique involves only 3 public matrices (one more matrix is required for explicit traceability to open the group signature) in group public-key, that is, $\text{Gpk} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3)$ where $\mathbf{A}_0, \mathbf{A}_j^j \in \mathbb{Z}_q^{n \times m}$. To generate the secret-key for group member $i \in \{1, 2, \dots, N\}$, it only needs to define a different matrix $\mathbf{A}_i = [\mathbf{A}_0 | \mathbf{A}_1^1 + i\mathbf{A}_2^2] \in \mathbb{Z}_q^{n \times 2m}$, and the secret-key of i is a short trapdoor basis of a classical q -ary lattice $\Lambda_q^\perp(\mathbf{A}_i)$. Here, the group size $N < q$ is required for the collision resistance, and this can be done simply (just setting q , a polynomial in security parameter n , big enough).

Thus the above identity-encoding technique provides the following 2 benefits:

- It just needs 3 public matrices for identity-encoding, thus this construction provides a shorter group public-key compared with other schemes which have the number of public matrices at least $\mathcal{O}(\log N)$.
- It shows a simple group membership relation allowing to construct the efficient statistical ZKP protocol for above relation.

3 Preparations

This section describes the main techniques, also serve as the main building blocks that will be used in our new design of Stern-type statistical ZKP protocol.

3.1 The Improved of Identity-Encoding Technique

A public vector $\mathbf{u} \leftarrow_{\mathcal{S}} \mathbb{Z}_q^n$ is required, i.e., $\text{Gpk} = (\mathbf{A}_0, \mathbf{A}_1^1, \mathbf{A}_2^2, \mathbf{A}_3^3, \mathbf{u})$, furthermore, the secret-key of i is not yet a trapdoor basis matrix for $\Lambda_q^\perp(\mathbf{A}_i)$, instead of a short $2m$ -dimensional vector $\mathbf{e}_i = (\mathbf{e}_{i,1}, \mathbf{e}_{i,2}) \in \mathbb{Z}^{2m}$ in the coset of $\Lambda_q^\perp(\mathbf{A}_i)$, i.e., $\Lambda_q^\perp(\mathbf{A}_i) = \{\mathbf{e}_i \in \mathbb{Z}^{2m} \mid \mathbf{A}_i \cdot \mathbf{e}_i = \mathbf{u} \bmod q\}$.

In order to design a new and efficient Stern-type statistical ZKP protocol corresponding to the above variant, we need to transform identity-encoding matrix $\mathbf{A}_i = [\mathbf{A}_0 | \mathbf{A}_1^1 + i\mathbf{A}_2^2]$ of member i to a new form. Before do that, we first define two notations (we restate, in this paper, the group size $N = 2^\ell$):

- $\mathbf{g}_\ell = (1, 2^1, 2^2, \dots, 2^{\ell-1})$: a power-of-2 vector, for an integer $i \in \{1, 2, \dots, N\}$, $i = \mathbf{g}_\ell^\top \cdot \text{bin}(i)$ where $\text{bin}(i) \in \{0, 1\}^\ell$ denotes a binary representation of i .
- \otimes : a concatenation with vectors or matrices, given $\mathbf{e} = (e_1, e_2, \dots, e_\ell) \in \mathbb{Z}_q^\ell$, $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{e}' \in \mathbb{Z}_q^m$, define:

$$\mathbf{e} \otimes \mathbf{e}' = (e_1\mathbf{e}', e_2\mathbf{e}', \dots, e_\ell\mathbf{e}') \in \mathbb{Z}_q^{m\ell}, \quad \mathbf{e} \otimes \mathbf{A} = [e_1\mathbf{A} | e_2\mathbf{A} | \dots | e_\ell\mathbf{A}] \in \mathbb{Z}_q^{n \times m\ell}.$$

Next, we transform \mathbf{A}_i to \mathbf{A}' that is independent of the index of member i , where $\mathbf{A}' = [\mathbf{A}_0 | \mathbf{A}_1^1 | \mathbf{A}_2^2 | \dots | 2^{\ell-1}\mathbf{A}_2^2] = [\mathbf{A}_0 | \mathbf{A}_1^1 | \mathbf{g}_\ell \otimes \mathbf{A}_2^2] \in \mathbb{Z}_q^{n \times (\ell+2)m}$.

As a corresponding revision to the secret-key of member i , $\mathbf{e}_i = (\mathbf{e}_{i,1}, \mathbf{e}_{i,2})$ is transformed to \mathbf{e}'_i , a vector with a special structure, $\mathbf{e}'_i = (\mathbf{e}_{i,1}, \mathbf{e}_{i,2}, \text{bin}(i) \otimes \mathbf{e}_{i,2}) \in \mathbb{Z}^{(\ell+2)m}$.

Thus from the above transformations, the relation $\mathbf{A}_i \cdot \mathbf{e}_i = \mathbf{u} \bmod q$ is now transformed to the following new form,

$$\mathbf{A}_i \cdot \mathbf{e}_i = \mathbf{A}' \cdot \mathbf{e}'_i = \mathbf{u} \bmod q. \quad (1)$$

3.2 The New Creation of Revocation Token

The revocation token of member i in our new design is constructed by \mathbf{A}_0 and a short Gaussian vector $\mathbf{r} \in \mathbb{Z}^m$, that is, $\text{grt}_i = \mathbf{A}_0 \cdot \mathbf{r} \bmod q$, which is separating from the group member secret-key. Therefore the underlying improved lattice-based GS-VLR scheme can obtain a stronger security, *almost-full anonymity*, first defined in [27].

For the revocation mechanism, as it was stated in [19], due to a flaw in the revocation mechanism of [15] which adopts the *inequality test* method to check whether the signer's revocation token belongs to a given revocation list or not, a corrected technique which realizes revocation by binding signer's revocation token grt_i to an LWE function was proposed,

$$\mathbf{b} = \mathbf{B}^\top \text{grt}_i + \mathbf{e}_0 = (\mathbf{B}^\top \mathbf{A}_0) \cdot \mathbf{r} + \mathbf{e}_0 \bmod q, \quad (2)$$

where $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ is a random matrix from a random oracle and vector $\mathbf{e}_0 \in \mathbb{Z}^m$ is sampled from the LWE error χ^m .

3.3 The Explicit Traceability Mechanism

For the explicit traceability mechanism, as it was shown in [21], the dual LWE cryptosystem [10] can be used to hide the identity index of signer i . In our new design, the binary string $\text{bin}(i) \in \{0, 1\}^\ell$ is treated as plaintext in the public-key encryption cryptosystem, and the ciphertext can be expressed as

$$\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) = (\mathbf{A}_3^{3\top} \mathbf{s} + \mathbf{e}_1, \mathbf{G}^\top \mathbf{s} + \mathbf{e}_2 + \lfloor q/2 \rfloor \text{bin}(i)) \bmod q$$

where $\mathbf{G} \in \mathbb{Z}_q^{n \times \ell}$ is a random matrix, and \mathbf{s} , \mathbf{e}_1 , \mathbf{e}_2 are random vectors sampled from the LWE error χ^n , χ^m , χ^ℓ , respectively.

Thus, the above relation can be expressed as:

$$\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) = \mathbf{P}\mathbf{e} + (\mathbf{0}^m, \lfloor q/2 \rfloor \text{bin}(i)) \bmod q, \quad (3)$$

where $\mathbf{P} = \left(\begin{array}{c|c} \mathbf{A}_3^{3\top} & \\ \cdots & \mathbf{I}_{m+\ell} \\ \mathbf{G}^\top & \end{array} \right) \in \mathbb{Z}_q^{(m+\ell) \times (n+m+\ell)}$, and $\mathbf{e} = \begin{pmatrix} \mathbf{s} \\ \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \in \mathbb{Z}^{n+m+\ell}$.

Taking all the above transformations ideas and the versatility of the Stern-extension argument system introduced by Ling et al. [23] together, we can design a new and efficient Stern-type statistical ZKP protocol to prove the above new relations (1), (2) and (3).

4 A New Stern-Type Zero-Knowledge Proofs Protocol

A new and efficient Stern-type statistical ZKP protocol that allows the signer \mathcal{P} to convince the verifier \mathcal{V} that \mathcal{P} is indeed a group member who honestly signed the message $M \in \{0, 1\}^*$ will be introduced, i.e., \mathcal{P} owns a valid group member secret-key, its revocation token is correctly embedded into an LWE instance, and its identity information, a binary representation of its index is correctly hidden within the dual LWE cryptosystem.

Firstly, we define some specific sets and techniques as in [14, 15, 19]. Given a binary string $\text{id} = (d_1, d_2, \dots, d_\ell) \in \{0, 1\}^\ell$, we define:

1. $\mathbf{B}_{2\ell}$: the set of all vectors in $\{0, 1\}^{2\ell}$ having the *Hamming weight* ℓ .
2. \mathbf{B}_{3m} : the set of all vectors in $\{-1, 0, 1\}^{3m}$ having same number of -1 , 0 , and 1 , that is, m coordinates -1 , m coordinates 1 , and m coordinates 0 .
3. $\text{Sec}_\beta(\text{id})$: the set of all vectors having some specific structure and norm, i.e., $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, d_1 \mathbf{e}_2, \dots, d_\ell \mathbf{e}_2) \in \mathbb{Z}_q^{(\ell+2)m}$, and $\|\mathbf{e}\|_\infty \leq \beta$.
4. $\text{SecExt}(\text{id}^*)$: the set of all vectors having some specific structure for $\text{id}^* \in \mathbf{B}_{2\ell}$, an extension of id , i.e., $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2, d_1 \mathbf{e}_2, d_2 \mathbf{e}_2, \dots, d_\ell \mathbf{e}_2) \in \{-1, 0, 1\}^{(2\ell+2)3m}$, and $\mathbf{e}_1, \mathbf{e}_2 \in \mathbf{B}_{3m}$.

Given $\mathbf{e} = (\mathbf{e}_{-1}, \mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_{2\ell}) \in \mathbb{Z}_q^{(2\ell+2)3m}$ and 3 permutations $\pi, \varphi \in \mathcal{S}_{3m}$, $\tau \in \mathcal{S}_{2\ell}$, we define a composition \mathcal{T} as follows:

$$\mathcal{T}_{\pi, \varphi, \tau}(\mathbf{e}) = (\pi(\mathbf{e}_{-1}), \varphi(\mathbf{e}_0), \varphi(\mathbf{e}_{\tau(1)}), \varphi(\mathbf{e}_{\tau(2)}) \cdots, \varphi(\mathbf{e}_{\tau(2\ell)})).$$

In particular, given $\text{id} \in \{0, 1\}^\ell$, $\pi, \varphi \in \mathcal{S}_{3m}$, $\tau \in \mathcal{S}_{2\ell}$, $\mathbf{e} \in \mathbb{Z}_q^{(2\ell+2)3m}$, it can be checked that $\mathbf{e} \in \text{SecExt}(\text{id}^*) \Leftrightarrow \mathcal{T}_{\pi, \varphi, \tau}(\mathbf{e}) \in \text{SecExt}(\tau(\text{id}^*))$, here $\text{id}^* \in \mathbb{B}_{2\ell}$ is an extension of $\text{id} \in \{0, 1\}^\ell$.

Secondly, we recall the Decomposition-Extension (Dec-Ext) and Matrix Extension (Matrix-Ext) techniques which were first introduced in [15].

Let $k = \lfloor \log \beta \rfloor + 1$, and define a sequence of integers,

$$\beta_1 = \lceil \beta/2 \rceil, \beta_2 = \lceil (\beta - \beta_1)/2 \rceil, \beta_3 = \lceil (\beta - \beta_1 - \beta_2)/2 \rceil, \dots, \beta_k = 1.$$

Dec: Given $\mathbf{e} = (e_1, e_2, \dots, e_m) \in \mathbb{Z}^m$, $\|\mathbf{e}\|_\infty \leq \beta$, the goal is to represent it by k vectors in $\{-1, 0, 1\}^m$. The procedure Dec works as follows:

1. For $i \in \{1, 2, \dots, m\}$, express e_i as $\sum_{j=1}^k \beta_j e_{i,j}$, where $e_{i,j} \in \{-1, 0, 1\}$.
2. For $j \in \{1, 2, \dots, k\}$, define $\hat{\mathbf{e}}_j = (e_{1,j}, e_{2,j}, \dots, e_{m,j})$. Thus, we have $\hat{\mathbf{e}}_j \in \{-1, 0, 1\}^m$ and $\mathbf{e} = \sum_{j=1}^k \beta_j \hat{\mathbf{e}}_j$.

Ext: Given $\hat{\mathbf{e}}_j \in \{-1, 0, 1\}^m$, the goal is to extend it to $\mathbf{e}_j \in \mathbb{B}_{3m}$. The procedure Ext works as follows:

1. Let the numbers of different coordinates -1 , 0 and 1 in $\hat{\mathbf{e}}_j$ are λ_{-1} , λ_0 and λ_1 , respectively.
2. Choose $\mathbf{e}'_j \in \{-1, 0, 1\}^{2m}$ which has the numbers of different coordinates -1 , 0 and 1 exactly $(m - \lambda_{-1})$, $(m - \lambda_0)$ and $(m - \lambda_1)$, respectively.
3. Let $\mathbf{e}_j = (\hat{\mathbf{e}}_j, \mathbf{e}'_j) \in \{-1, 0, 1\}^{3m}$. Thus for $\pi \in \mathcal{S}_{3m}$, $\mathbf{e} \in \mathbb{B}_{3m} \Leftrightarrow \pi(\mathbf{e}) \in \mathbb{B}_{3m}$.

Matrix-Ext: Given $\mathbf{A}' = [\mathbf{A} | \mathbf{A}_0 | \dots | \mathbf{A}_\ell] \in \mathbb{Z}_q^{n \times (\ell+2)m}$, the goal is to extend it to $\mathbf{A}^* \in \mathbb{Z}_q^{n \times (2\ell+2)3m}$. The procedure Matrix-Ext works as follows:

1. Add $\mathbf{0}^{n \times 2m}$ to each of component-matrices and ℓ blocks of $\mathbf{0}^{n \times 3m}$.
2. Output $\mathbf{A}^* = [\mathbf{A} | \mathbf{0}^{n \times 2m} | \mathbf{A}_0 | \mathbf{0}^{n \times 2m} | \dots | \mathbf{A}_\ell | \mathbf{0}^{n \times 2m} | \mathbf{0}^{n \times 3m \ell}]$.

In the following contents, we introduce our main contribution, a new Stern-type statistical ZKP protocol for improved lattice-based GS-VLR enjoying those three significant advantages mentioned previously.

The new Stern-type statistical ZKP protocol between \mathcal{P} and \mathcal{V} can be summarized as follows:

1. The public inputs are $\mathbf{A}' = [\mathbf{A}_0 | \mathbf{A}_1^1 | \mathbf{g}_\ell \otimes \mathbf{A}_2^2] \in \mathbb{Z}_q^{n \times (\ell+2)m}$, $\mathbf{b} \in \mathbb{Z}_q^m$, $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{P} = \left(\begin{array}{c|c} \mathbf{A}_3^{3\top} & \\ \cdots & \mathbf{I}_{m+\ell} \\ \mathbf{G}^\top & \end{array} \right) \in \mathbb{Z}_q^{(m+\ell) \times (n+m+\ell)}$ and $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$.

2. \mathcal{P} 's witnesses are $\mathbf{e}' = (\mathbf{e}'_1, \mathbf{e}'_2, \text{bin}(i) \otimes \mathbf{e}'_2) \in \text{Sec}_\beta(\text{id})$ for a secret index $i \in \{1, 2, \dots, N\}$, and three short vectors \mathbf{r} , $\mathbf{e}_0 \in \chi^m$ and $\mathbf{e} = \begin{pmatrix} \mathbf{s} \\ \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \in \mathbb{Z}^{n+m+\ell}$,

where $\mathbf{s} \in \chi^n$, $\mathbf{e}_1 \in \chi^m$, $\mathbf{e}_2 \in \chi^\ell$, the LWE errors.

3. \mathcal{P} 's goal is to convince \mathcal{V} in zero-knowledge that:

- 3.1. $\mathbf{A}' \cdot \mathbf{e}' = \mathbf{u} \bmod q$, where $\mathbf{e}' \in \text{Sec}_\beta(\text{id})$ and keeping $\text{id} \in \{0, 1\}^\ell$ secret.
- 3.2. $\mathbf{b} = (\mathbf{B}^\top \mathbf{A}_0) \cdot \mathbf{r} + \mathbf{e}_0 \bmod q$ where $\|\mathbf{r}\|_\infty, \|\mathbf{e}_0\|_\infty \leq \beta$.
- 3.3. $\mathbf{c} = \mathbf{P}\mathbf{e} + (\mathbf{0}^m, \lfloor q/2 \rfloor \text{bin}(i)) \bmod q$ where $\|\mathbf{e}\|_\infty \leq \beta$ and keeping $\text{bin}(i) \in \{0, 1\}^\ell$ secret.

Firstly, we sketch the Group Membership Mechanism, that is, \mathcal{P} is a certified group member and its goal is shown as in 3.1.

1. Parse $\mathbf{A}' = [\mathbf{A}_0 | \mathbf{A}_1^1 | \mathbf{A}_2^2 | 2\mathbf{A}_2^2] \cdots | 2^{\ell-1} \mathbf{A}_2^2]$, then use Matrix-Ext technique to extend it to $\mathbf{A}^* = [\mathbf{A}_0 | \mathbf{0}^{n \times 2m} | \mathbf{A}_1^1 | \mathbf{0}^{n \times 2m} | \cdots | 2^{\ell-1} \mathbf{A}_2^2 | \mathbf{0}^{n \times 2m} | \mathbf{0}^{n \times 3m\ell}]$
2. Parse $\text{id} = (d_1, \dots, d_\ell) \in \{0, 1\}^\ell$ and extend it to $\text{id}^* = (d_1, \dots, d_\ell, \dots, d_{2\ell}) \in \mathbf{B}_{2\ell}$.
3. Parse $\mathbf{e}' = (\mathbf{e}'_1, \mathbf{e}'_2, d_1 \mathbf{e}'_2, d_2 \mathbf{e}'_2, \dots, d_\ell \mathbf{e}'_2)$, use Dec, Ext techniques extending \mathbf{e}'_1 to k vectors $\mathbf{e}'_{1,1}, \mathbf{e}'_{1,2}, \dots, \mathbf{e}'_{1,k} \in \mathbf{B}_{3m}$, \mathbf{e}'_2 to k vectors $\mathbf{e}'_{2,1}, \mathbf{e}'_{2,2}, \dots, \mathbf{e}'_{2,k} \in \mathbf{B}_{3m}$, respectively. For each $j \in \{1, 2, \dots, k\}$, we define a new vector $\mathbf{e}'_j = (\mathbf{e}'_{1,j}, \mathbf{e}'_{2,j}, d_1 \mathbf{e}'_{2,j}, d_2 \mathbf{e}'_{2,j}, \dots, d_{2\ell} \mathbf{e}'_{2,j})$, it can be checked that $\mathbf{e}'_j \in \text{SecExt}(\text{id}^*)$.

Thus, \mathcal{P} 's goal in 3.1 is transformed to a new structure,

$$\mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{e}'_j) = \mathbf{u} \bmod q, \mathbf{e}'_j \in \text{SecExt}(\text{id}^*). \quad (4)$$

To prove the new relation (4) in zero-knowledge, we take 2 steps as follows:

1. Pick k random vectors $\mathbf{r}'_1, \dots, \mathbf{r}'_k \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+2)3m}$ to mask $\mathbf{e}'_1, \dots, \mathbf{e}'_k$, then it can be checked that $\mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j (\mathbf{e}'_j + \mathbf{r}'_j)) - \mathbf{u} = \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{r}'_j) \bmod q$.
2. Pick 2 permutations $\pi, \varphi \in \mathcal{S}_{3m}$, one permutation $\tau \in \mathcal{S}_{2\ell}$, then it can be checked that $\forall j \in \{1, 2, \dots, k\}$, $\mathcal{T}_{\pi, \varphi, \tau}(\mathbf{e}'_j) \in \text{SecExt}(\tau(\text{id}^*))$, where $\text{id}^* \in \mathbf{B}_{2\ell}$ is an extension of $\text{id} = \text{bin}(i) \in \{0, 1\}^\ell$.

Secondly, we sketch the Revocation Mechanism, that is, \mathcal{P} 's revocation token is correctly embedded in an LWE function and its goal is shown as in 3.2.

1. Let $\mathbf{B}' = \mathbf{B}^\top \mathbf{A}_0 \bmod q \in \mathbb{Z}_q^{m \times m}$.
2. Parse $\mathbf{r} = (r_1, r_2, \dots, r_m) \in \mathbb{Z}^m$, use Dec and Ext techniques to extend \mathbf{r} to k vectors $\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \dots, \mathbf{r}^{(k)} \in \mathbf{B}_{3m}$.
3. Parse $\mathbf{e}_0 = (e_1^0, e_2^0, \dots, e_m^0) \in \mathbb{Z}^m$, use Dec and Ext techniques to extend \mathbf{e}_0 to k vectors $\mathbf{e}_1^0, \mathbf{e}_2^0, \dots, \mathbf{e}_k^0 \in \mathbf{B}_{3m}$.
4. Let $\mathbf{B}^* = [\mathbf{B}' | \mathbf{0}^{n \times 2m} | \mathbf{I}_m | \mathbf{0}^{n \times 2m}]$, where \mathbf{I}_m is the identity matrix of order m .

Thus, \mathcal{P} 's goal in 3.2 is transformed to a new structure,

$$\mathbf{b} = \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{r}^{(j)} \\ \mathbf{e}_j^0 \end{pmatrix}) \bmod q, \mathbf{r}^{(j)}, \mathbf{e}_j^0 \in \mathbf{B}_{3m}. \quad (5)$$

To prove the new relation (5) in zero-knowledge, we take 2 steps as follows:

1. Pick k uniformly random vectors $\mathbf{r}_1, \dots, \mathbf{r}_k \xleftarrow{\$} \mathbb{Z}_q^{3m}$ to mask $\mathbf{r}^{(1)}, \dots, \mathbf{r}^{(k)}$.

2. Pick k random vectors $\mathbf{r}_1^0, \dots, \mathbf{r}_k^0 \xleftarrow{\$} \mathbb{Z}_q^{3m}$ to mask $\mathbf{e}_1^0, \dots, \mathbf{e}_k^0$, it can be checked that $\mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{r}^{(j)} + \mathbf{r}_j \\ \mathbf{e}_j^0 + \mathbf{r}_j^0 \end{pmatrix}) - \mathbf{b} = \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{r}_j \\ \mathbf{r}_j^0 \end{pmatrix}) \pmod q$.
3. Pick 2 permutations $\xi, \phi \in \mathcal{S}_{3m}$, then it can be checked that,

$$\forall j \in \{1, 2, \dots, k\}, \xi(\mathbf{r}^{(j)}), \phi(\mathbf{e}_j^0) \in \mathbf{B}_{3m}.$$

Thirdly, we sketch the Explicit Traceability Mechanism, i.e., \mathcal{P} 's identity index is correctly hidden in a dual LWE cryptosystem and its goal is shown as in 3.3.

1. Let $\mathbf{P}^* = [\mathbf{P} | \mathbf{0}^{(m+\ell) \times 2(n+m+\ell)}]$.
2. Let $\mathbf{Q} = \left(\begin{array}{c|c} \mathbf{0}^{m \times \ell} & \mathbf{0}^{m \times \ell} \\ \hline \dots & \dots \\ \lfloor q/2 \rfloor \mathbf{I}_\ell & \mathbf{0}^{\ell \times \ell} \end{array} \right)$, where \mathbf{I}_ℓ is the identity matrix of order ℓ .
3. Parse $\mathbf{e} = \begin{pmatrix} \mathbf{s} \\ \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \in \mathbb{Z}^{n+m+\ell}$, use Dec, Ext techniques to extend \mathbf{e} to k vectors $\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \dots, \mathbf{e}^{(k)} \in \mathbf{B}_{3(n+m+\ell)}$.
4. Let $\text{id}^* = \text{bin}(i)^* \in \mathbf{B}_{2\ell}$ be an extension of $\text{id} = \text{bin}(i)$.

Thus, \mathcal{P} 's goal in 3.3 is transformed to a new structure,

$$\mathbf{c} = \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{e}^{(j)}) + \mathbf{Q} \cdot \text{id}^* \pmod q, \quad \mathbf{e}^{(j)} \in \mathbf{B}_{3(n+m+\ell)}, \text{id}^* \in \mathbf{B}_{2\ell}. \quad (6)$$

To prove the new relation (6) in zero-knowledge, we take 2 steps as follows:

1. Pick a random vector $\mathbf{r}_{\text{id}^*} \xleftarrow{\$} \mathbb{Z}_q^{2\ell}$ to mask $\text{id}^* = \text{bin}(i)^*$.
2. Pick k random vectors $\mathbf{r}_1'', \dots, \mathbf{r}_k'' \xleftarrow{\$} \mathbb{Z}_q^{3(n+m+\ell)}$ to mask $\mathbf{e}^{(1)}, \dots, \mathbf{e}^{(k)}$, it can be checked that,

$$\mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j (\mathbf{e}^{(j)} + \mathbf{r}_j'')) + \mathbf{Q} \cdot (\text{id}^* + \mathbf{r}_{\text{id}^*}) - \mathbf{c} = \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{r}_j'') + \mathbf{Q} \cdot \mathbf{r}_{\text{id}^*} \pmod q.$$

3. Pick one permutation $\rho \in \mathcal{S}_{3(n+m+\ell)}$, it can be checked that,

$$\forall j \in \{1, 2, \dots, k\}, \rho(\mathbf{e}^{(j)}) \in \mathbf{B}_{3(n+m+\ell)}, \tau(\text{id}^*) \in \mathbf{B}_{2\ell},$$

where τ has been picked in the proof of Group Membership Mechanism.

Putting the above techniques together, we obtain a new Stern-type interactive statistical ZKP protocol, and the details will be given bellow.

In our design, we also utilize a statistically hiding and computationally blinding commitment scheme (COM) as proposed in [12]. For simplicity, we omit the randomness of COM.

The prover \mathcal{P} and verifier \mathcal{V} interact as follows:

1. Commitments: \mathcal{P} first randomly samples the following random objects:

$$\begin{cases} \mathbf{r}'_1, \dots, \mathbf{r}'_k \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+2)3m}; \mathbf{r}_1, \dots, \mathbf{r}_k, \mathbf{r}_1^0, \dots, \mathbf{r}_k^0 \xleftarrow{\$} \mathbb{Z}_q^{3m}; \mathbf{r}_{\text{id}^*} \xleftarrow{\$} \mathbb{Z}_q^{2\ell}; \\ \mathbf{r}''_1, \dots, \mathbf{r}''_k \xleftarrow{\$} \mathbb{Z}_q^{3(n+m+\ell)}; \pi_1, \dots, \pi_k \xleftarrow{\$} \mathcal{S}_{3m}; \varphi_1, \dots, \varphi_k \xleftarrow{\$} \mathcal{S}_{3m}; \\ \rho_1, \dots, \rho_k \xleftarrow{\$} \mathcal{S}_{3(n+m+\ell)}; \xi_1, \dots, \xi_k, \phi_1, \dots, \phi_k \xleftarrow{\$} \mathcal{S}_{3m}; \tau \xleftarrow{\$} \mathcal{S}_{2\ell}. \end{cases}$$

\mathcal{P} sends the commitment $\text{CMT} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ to \mathcal{V} , where

$$\begin{cases} \mathbf{c}_1 = \text{COM}(\{\pi_j, \varphi_j, \xi_j, \phi_j, \rho_j\}_{j=1}^k, \tau, \mathbf{A}^* (\sum_{j=1}^k \beta_j \mathbf{r}'_j), \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{r}_j \\ \mathbf{r}_j^0 \end{pmatrix}), \\ \quad \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{r}''_j) + \mathbf{Q} \cdot \mathbf{r}_{\text{id}^*}), \\ \mathbf{c}_2 = \text{COM}(\{\mathcal{T}_{\pi_j, \varphi_j, \tau}(\mathbf{r}'_j), \xi_j(\mathbf{r}_j), \phi_j(\mathbf{r}_j^0), \rho_j(\mathbf{r}''_j)\}_{j=1}^k, \tau(\mathbf{r}_{\text{id}^*})), \\ \mathbf{c}_3 = \text{COM}(\{\mathcal{T}_{\pi_j, \varphi_j, \tau}(\mathbf{e}'_j + \mathbf{r}'_j), \xi_j(\mathbf{r}^{(j)} + \mathbf{r}_j), \phi_j(\mathbf{e}_j^0 + \mathbf{r}_j^0), \rho_j(\mathbf{e}^{(j)} + \mathbf{r}''_j)\}_{j=1}^k, \\ \quad \tau(\text{id}^* + \mathbf{r}_{\text{id}^*})). \end{cases}$$

2. Challenge: \mathcal{V} chooses a challenge $\text{CH} \xleftarrow{\$} \{1, 2, 3\}$ and sends it to \mathcal{P} .

3. Response: Depending on CH, \mathcal{P} replies as follows:

◦ If CH = 1. For $j \in \{1, 2, \dots, k\}$, let $\mathbf{v}'_j = \mathcal{T}_{\pi_j, \varphi_j, \tau}(\mathbf{e}'_j)$, $\mathbf{w}'_j = \mathcal{T}_{\pi_j, \varphi_j, \tau}(\mathbf{r}'_j)$, $\mathbf{v}_j = \xi_j(\mathbf{r}^{(j)})$, $\mathbf{w}_j = \xi_j(\mathbf{r}_j)$, $\mathbf{v}_j^0 = \phi_j(\mathbf{e}_j^0)$, $\mathbf{w}_j^0 = \phi_j(\mathbf{r}_j^0)$, $\mathbf{v}^{(j)} = \rho_j(\mathbf{e}^{(j)})$, $\mathbf{w}''_j = \rho_j(\mathbf{r}''_j)$, $\mathbf{t}_{\text{id}} = \tau(\text{id}^*)$ and $\mathbf{v}_{\text{id}} = \tau(\mathbf{r}_{\text{id}^*})$, define

$$\text{RSP} = (\{\mathbf{v}'_j, \mathbf{w}'_j, \mathbf{v}_j, \mathbf{w}_j, \mathbf{v}_j^0, \mathbf{w}_j^0, \mathbf{v}^{(j)}, \mathbf{w}''_j\}_{j=1}^k, \mathbf{t}_{\text{id}}, \mathbf{v}_{\text{id}}). \quad (7)$$

◦ If CH = 2. For $j \in \{1, 2, \dots, k\}$, let $\hat{\pi}_j = \pi_j$, $\hat{\varphi}_j = \varphi_j$, $\hat{\xi}_j = \xi_j$, $\hat{\phi}_j = \phi_j$, $\hat{\rho}_j = \rho_j$, $\hat{\tau} = \tau$, $\mathbf{x}'_j = \mathbf{e}'_j + \mathbf{r}'_j$, $\mathbf{x}_j = \mathbf{r}^{(j)} + \mathbf{r}_j$, $\mathbf{x}_j^0 = \mathbf{e}_j^0 + \mathbf{r}_j^0$, $\mathbf{x}''_j = \mathbf{e}^{(j)} + \mathbf{r}''_j$ and $\mathbf{x}_{\text{id}} = \text{id}^* + \mathbf{r}_{\text{id}^*}$, define

$$\text{RSP} = (\{\hat{\pi}_j, \hat{\varphi}_j, \hat{\xi}_j, \hat{\phi}_j, \hat{\rho}_j, \mathbf{x}'_j, \mathbf{x}_j, \mathbf{x}_j^0, \mathbf{x}''_j\}_{j=1}^k, \hat{\tau}, \mathbf{x}_{\text{id}}). \quad (8)$$

◦ If CH = 3. For $j \in \{1, 2, \dots, k\}$, let $\tilde{\pi}_j = \pi_j$, $\tilde{\varphi}_j = \varphi_j$, $\tilde{\xi}_j = \xi_j$, $\tilde{\phi}_j = \phi_j$, $\tilde{\rho}_j = \rho_j$, $\tilde{\tau} = \tau$, $\mathbf{h}'_j = \mathbf{r}'_j$, $\mathbf{h}_j = \mathbf{r}_j$, $\mathbf{h}_j^0 = \mathbf{r}_j^0$, $\mathbf{h}''_j = \mathbf{r}''_j$ and $\mathbf{h}_{\text{id}} = \mathbf{r}_{\text{id}^*}$, define

$$\text{RSP} = (\{\tilde{\pi}_j, \tilde{\varphi}_j, \tilde{\xi}_j, \tilde{\phi}_j, \tilde{\rho}_j, \mathbf{h}'_j, \mathbf{h}_j, \mathbf{h}_j^0, \mathbf{h}''_j\}_{j=1}^k, \tilde{\tau}, \mathbf{h}_{\text{id}}). \quad (9)$$

4. Verification: Receiving RSP, \mathcal{V} checks as follows:

◦ If CH = 1. Check that $\mathbf{t}_{\text{id}} \in \mathcal{B}_{2\ell}$, for $j \in \{1, 2, \dots, k\}$, $\mathbf{v}'_j \in \text{SecExt}(\mathbf{t}_{\text{id}})$, $\mathbf{v}_j \in \mathcal{B}_{3m}$, $\mathbf{v}^{(j)} \in \mathcal{B}_{3(n+m+\ell)}$, $\mathbf{v}_j^0 \in \mathcal{B}_{3m}$ and that,

$$\begin{cases} \mathbf{c}_2 = \text{COM}(\{\mathbf{w}'_j, \mathbf{w}_j, \mathbf{w}_j^0, \mathbf{w}''_j\}_{j=1}^k, \mathbf{v}_{\text{id}}), \\ \mathbf{c}_3 = \text{COM}(\{\mathbf{v}'_j + \mathbf{w}'_j, \mathbf{v}_j + \mathbf{w}_j, \mathbf{v}_j^0 + \mathbf{w}_j^0, \mathbf{v}^{(j)} + \mathbf{w}''_j\}_{j=1}^k, \mathbf{t}_{\text{id}} + \mathbf{v}_{\text{id}}). \end{cases}$$

◦ If $\text{CH} = 2$. For $j \in \{1, 2, \dots, k\}$, check that,

$$\left\{ \begin{array}{l} \dot{\mathbf{c}}_1 = \text{COM}(\{\hat{\pi}_j, \hat{\varphi}_j, \hat{\xi}_j, \hat{\phi}_j, \hat{\rho}_j\}_{j=1}^k, \hat{\tau}, \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{x}'_j) - \mathbf{u}, \\ \quad \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{x}_j \\ \mathbf{x}_j^0 \end{pmatrix}) - \mathbf{b}, \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{x}''_j) + \mathbf{Q}^* \cdot \mathbf{x}_{\text{id}} - \mathbf{c}), \\ \dot{\mathbf{c}}_3 = \text{COM}(\{\mathcal{T}_{\hat{\pi}_j, \hat{\varphi}_j, \hat{\tau}}(\mathbf{x}'_j), \hat{\xi}_j(\mathbf{x}_j), \hat{\phi}_j(\mathbf{x}_j^0), \hat{\rho}_j(\mathbf{x}''_j)\}_{j=1}^k, \hat{\tau}(\mathbf{x}_{\text{id}})). \end{array} \right.$$

◦ If $\text{CH} = 3$. For $j \in \{1, 2, \dots, k\}$, check that,

$$\left\{ \begin{array}{l} \dot{\mathbf{c}}_1 = \text{COM}(\{\tilde{\pi}_j, \tilde{\varphi}_j, \tilde{\xi}_j, \tilde{\phi}_j, \tilde{\rho}_j\}_{j=1}^k, \tilde{\tau}, \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{h}'_j), \\ \quad \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{h}_j \\ \mathbf{h}_j^0 \end{pmatrix}), \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{h}''_j) + \mathbf{Q}^* \cdot \mathbf{h}_{\text{id}}), \\ \dot{\mathbf{c}}_2 = \text{COM}(\{\mathcal{T}_{\tilde{\pi}_j, \tilde{\varphi}_j, \tilde{\tau}}(\mathbf{h}'_j), \tilde{\xi}_j(\mathbf{h}_j), \tilde{\phi}_j(\mathbf{h}_j^0), \tilde{\rho}_j(\mathbf{h}''_j)\}_{j=1}^k, \tilde{\tau}(\mathbf{h}_{\text{id}})). \end{array} \right.$$

The verifier \mathcal{V} outputs 1 iff all the above conditions hold, otherwise 0.

Thus, the associated relation $\mathcal{R}(n, k, \ell, q, m, \beta)$ in the above protocol can be defined as:

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{A}_0, \mathbf{A}_1^1, \mathbf{A}_2^2, \mathbf{B} \in \mathbb{Z}_q^{n \times m}, \mathbf{P} \in \mathbb{Z}_q^{(m+\ell) \times (n+m+\ell)}, \mathbf{u} \in \mathbb{Z}_q^n, \mathbf{b} \in \mathbb{Z}_q^m, \mathbf{c} \in \mathbb{Z}_q^{m+\ell}, \\ \text{id} = \text{bin}(i) \in \{0, 1\}^\ell, \mathbf{e}' = (\mathbf{e}'_1, \mathbf{e}'_2, \text{bin}(i) \otimes \mathbf{e}'_2) \in \text{Sec}_\beta(\text{id}), \mathbf{r}, \mathbf{e}_0 \in \mathbb{Z}^m, \\ \mathbf{e} = \begin{pmatrix} \mathbf{s} \\ \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \in \mathbb{Z}^{n+m+\ell}; \text{ s.t. } 0 < \|\mathbf{e}'\|_\infty, \|\mathbf{r}\|_\infty, \|\mathbf{e}_0\|_\infty, \|\mathbf{e}\|_\infty \leq \beta, \\ [\mathbf{A}_0 | \mathbf{A}_1^1 | \mathbf{g}_\ell \otimes \mathbf{A}_2^2] \cdot \mathbf{e}' = \mathbf{u} \text{ mod } q, \mathbf{b} = (\mathbf{B}^\top \mathbf{A}_0) \cdot \mathbf{r} + \mathbf{e}_0 \text{ mod } q, \\ \mathbf{c} = \mathbf{P}\mathbf{e} + (0^m, \lfloor q/2 \rfloor \text{bin}(i)) \text{ mod } q. \end{array} \right.$$

5 Analysis of the Proposed Protocol

The detailed analysis of the interactive protocol designed in Sect. 4 including 4 aspects: communication cost, perfect completeness, statistical zero-knowledge and argument of knowledge.

Theorem 1. *Let COM (as proposed in [12]) be a statistically hiding and computationally binding commitment scheme, thus for a given CMT, 3 valid responses RSP_1 , RSP_2 and RSP_3 with respect to 3 different challenges CH_1 , CH_2 and CH_3 , the proposed protocol is a statistical ZKAoK for $\mathcal{R}(n, k, \ell, q, m, \beta)$, where each round has perfect completeness, soundness error $2/3$, argument of knowledge property and communication cost $\tilde{\mathcal{O}}(\ell n \log \beta)$.*

Proof. The proof for this theorem will employ a list of standard proof techniques for Stern-type protocol as in [12, 14, 15], and it includes the following 4 aspects:

Communication Cost

- The output of COM, a vector of \mathbb{Z}_q^n , has bit-sizes $n \log q$, thus \mathcal{P} sends 3 commitments amounting to $3n \log q$ bits.
- The challenge $\text{CH} \in \{1, 2, 3\}$ could be represented by 2 bits.
- The response RSP from \mathcal{P} consist of the following items:
 - One permutation in $\mathcal{S}_{2\ell}$, $4k$ permutations in \mathcal{S}_{3m} ,
 - k permutations in $\mathcal{S}_{3(n+m+\ell)}$,
 - $2k$ vectors in $\mathbb{Z}_q^{(2\ell+2)3m}$, $2k$ vectors in $\mathbb{Z}_q^{3(n+m+\ell)}$,
 - $4k$ vectors in \mathbb{Z}_q^{3m} , one vector in $\{0, 1\}^{2\ell}$, one vector in $\mathbb{Z}_q^{2\ell}$.

Thus, the bit-size of RSP is bound by $\mathcal{O}(\ell mk) \log q$. Recall that $k = \lfloor \log \beta \rfloor + 1$, the communication cost of the proposed Stern-type statistical ZKP protocol is bounded by $\tilde{\mathcal{O}}(\ell n \log \beta)$.

Perfect Completeness

To show that given a tuple $(\mathbf{A}_0, \mathbf{A}_1^1, \mathbf{A}_2^2, \mathbf{P}, \mathbf{u}, \mathbf{B}, \mathbf{b}, \mathbf{c})$, if an honest prover \mathcal{P} owns a witness ($\text{id} = \text{bin}(i) \in \{0, 1\}^\ell$, $\mathbf{e}' \in \text{Sec}_\beta(\text{id})$, $\mathbf{r}, \mathbf{e}_0, \mathbf{e}_1 \in \mathbb{Z}^m$, $\mathbf{s} \in \mathbb{Z}^n$, $\mathbf{e}_2 \in \mathbb{Z}^\ell$), and follows the proposed protocol correctly, then \mathcal{P} can generate an efficient Stern-type statistical ZKP protocol satisfying the verification processes, and gets accepted by \mathcal{V} with a high probability.

Firstly, the public inputs and \mathcal{P} 's witness are transformed to \mathbf{A}^* , \mathbf{B}^* , \mathbf{P}^* , id^* and $\{\mathbf{e}'_j, \mathbf{r}^{(j)}, \mathbf{e}_j^0, \mathbf{e}^{(j)}\}_{j=1}^k$ by using the Dec, Ext and Matrix-Ext techniques, thus these new results satisfy the following new structures,

$$\begin{aligned} \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{e}'_j) &= \mathbf{u} \bmod q, \mathbf{e}'_j \in \text{SecExt}(\text{id}^*), \\ \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{r}^{(j)} \\ \mathbf{e}_j^0 \end{pmatrix}) &= \mathbf{b} \bmod q, \mathbf{r}^{(j)}, \mathbf{e}_j^0 \in \mathbf{B}_{3m}. \\ \mathbf{c} = \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{e}^{(j)}) + \mathbf{Q} \cdot \text{id}^* &\bmod q, \mathbf{e}^{(j)} \in \mathbf{B}_{3(n+m+\ell)}, \text{id}^* \in \mathbf{B}_{2\ell}. \end{aligned}$$

Next, to show that \mathcal{P} can correctly pass all the verification checks for each challenge $\text{CH} \in \{1, 2, 3\}$ with a high probability. Furthermore, apart from considering the checks for correct computations, it only needs to note that:

- If $\text{CH} = 1$. $\text{id} = \text{bin}(i) \in \{0, 1\}^\ell$, $\text{id}^* \in \mathbf{B}_{2\ell}$ is an extension of id , and $\mathbf{B}_{2\ell}$ is invariant under the permutation $\tau \in \mathcal{S}_{2\ell}$, thus we have $\mathbf{t}_{\text{id}} = \tau(\text{id}^*) \in \mathbf{B}_{2\ell}$. Similarly, for each $j \in \{1, \dots, k\}$, $\mathbf{r}^{(j)}, \mathbf{e}_j^0 \in \mathbf{B}_{3m}$, and \mathbf{B}_{3m} is invariant under $\xi_j, \phi_j \in \mathcal{S}_{3m}$, we have $\mathbf{v}_j = \xi_j(\mathbf{r}^{(j)}) \in \mathbf{B}_{3m}$ and $\mathbf{v}_j^0 = \phi_j(\mathbf{e}_j^0) \in \mathbf{B}_{3m}$; $\mathbf{e}^{(j)} \in \mathbf{B}_{3(n+m+\ell)}$, and $\mathbf{B}_{3(n+m+\ell)}$ is invariant under $\rho_j \in \mathcal{S}_{3(n+m+\ell)}$, thus we have $\mathbf{v}^{(j)} = \rho_j(\mathbf{e}^{(j)}) \in \mathbf{B}_{3(n+m+\ell)}$. As for \mathbf{e}'_j , we have

$$\mathbf{v}'_j = \mathcal{T}_{\pi_j, \varphi_j, \tau}(\mathbf{e}'_j) \in \text{SecExt}(\tau(\text{id}^*)) = \text{SecExt}(\mathbf{t}_{\text{id}}).$$

○ If $\text{CH} = 2$. The key point is to check \mathbf{c}'_1 , for $j \in \{1, 2, \dots, k\}$, \mathcal{P} can pass this step by generating $\mathbf{x}'_j, \mathbf{r}'_j, \mathbf{x}_j, \mathbf{r}_j, \mathbf{x}_j^0, \mathbf{r}_j^0, \mathbf{x}_j'', \mathbf{r}_j'', \mathbf{x}_{\text{id}}$, such that the followings hold true:

$$\begin{aligned} \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{x}'_j) - \mathbf{u} &= \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j (\mathbf{e}'_j + \mathbf{r}'_j)) - \mathbf{u} \\ &= \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{r}'_j) \pmod{q}. \\ \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{x}^{(j)} \\ \mathbf{x}_j^0 \end{pmatrix}) - \mathbf{b} &= \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{r}^{(j)} + \mathbf{r}_j \\ \mathbf{e}_j^0 + \mathbf{r}_j^0 \end{pmatrix}) - \mathbf{b} \\ &= \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{r}_j \\ \mathbf{r}_j^0 \end{pmatrix}) \pmod{q}, \\ \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{x}_j'') + \mathbf{Q}^* \cdot \mathbf{x}_{\text{id}} - \mathbf{c} &= \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j (\mathbf{e}^{(j)} + \mathbf{r}_j'')) + \mathbf{Q} \cdot (\text{id}^* + \mathbf{r}_{\text{id}}^*) - \mathbf{c} \\ &= \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{r}_j'') + \mathbf{Q} \cdot \mathbf{r}_{\text{id}}^* \pmod{q}. \end{aligned}$$

○ If $\text{CH} = 3$. It only needs to consider the checks for correct computations, and obviously these are true.

Statistical Zero-Knowledge

To design a PPT simulator \mathcal{S} who interacts with a verifier \mathcal{V}' (maybe dishonest) to output a simulated transcript that is statistically close to one generated by an honest prover \mathcal{P} in the real interaction with probability negligibly close to $2/3$. The construction is as follows:

\mathcal{S} picks a value $\widetilde{\text{CH}} \xleftarrow{\$} \{1, 2, 3\}$ as a prediction that \mathcal{V}' will not choose.

○ If $\widetilde{\text{CH}} = 1$. \mathcal{S} does as follows:

1. Use linear algebra algorithm to compute k vectors $\mathbf{e}_1'', \dots, \mathbf{e}_k'' \in \mathbb{Z}_q^{(2\ell+1)3m}$ such that $\mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{e}_j'') = \mathbf{u} \pmod{q}$.
2. Use linear algebra algorithm to compute k vectors $\mathbf{r}^{(1)}, \dots, \mathbf{r}^{(k)} \in \mathbb{Z}_q^{3m}$ and k vectors $\widehat{\mathbf{e}}_1, \dots, \widehat{\mathbf{e}}_k \in \mathbb{Z}_q^{3m}$ such that $\mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{r}^{(j)} \\ \widehat{\mathbf{e}}_j \end{pmatrix}) = \mathbf{b} \pmod{q}$.
3. Use linear algebra algorithm to compute k vectors $\mathbf{e}_1''', \dots, \mathbf{e}_k''' \in \mathbb{Z}_q^{3(n+m+\ell)}$ and $\text{id}^* \in \mathbb{Z}_q^{2\ell}$ such that $\mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{e}_j''') + \mathbf{Q} \cdot \text{id}^* = \mathbf{c} \pmod{q}$.
4. Sample several random vectors and permutations,

$$\left\{ \begin{array}{l} \mathbf{r}'_1, \dots, \mathbf{r}'_k \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+2)3m}; \mathbf{r}_1, \dots, \mathbf{r}_k; \mathbf{r}_1^0, \dots, \mathbf{r}_k^0 \xleftarrow{\$} \mathbb{Z}_q^{3m}; \mathbf{r}_{\text{id}}^* \xleftarrow{\$} \mathbb{Z}_q^{2\ell}; \\ \mathbf{r}''_1, \dots, \mathbf{r}''_k \xleftarrow{\$} \mathbb{Z}_q^{3(n+m+\ell)}; \pi_1, \dots, \pi_k \xleftarrow{\$} \mathcal{S}_{3m}; \varphi_1, \dots, \varphi_k \xleftarrow{\$} \mathcal{S}_{3m}; \\ \rho_1, \dots, \rho_k \xleftarrow{\$} \mathcal{S}_{3(n+m+\ell)}; \xi_1, \dots, \xi_k; \phi_1, \dots, \phi_k \xleftarrow{\$} \mathcal{S}_{3m}; \tau \xleftarrow{\$} \mathcal{S}_{2\ell}. \end{array} \right.$$

5. Compute $\text{CMT} = (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$, where

$$\left\{ \begin{array}{l} \mathbf{c}'_1 = \text{COM}(\{\pi_j, \varphi_j, \xi_j, \phi_j, \rho_j\}_{j=1}^k, \tau, \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{r}'_j), \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{r}_j \\ \mathbf{r}_j^0 \end{pmatrix}), \\ \quad \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{r}_j'') + \mathbf{Q} \cdot \mathbf{r}_{\text{id}}^*), \\ \mathbf{c}'_2 = \text{COM}(\{\mathcal{T}_{\pi_j, \varphi_j, \tau}(\mathbf{r}'_j), \xi_j(\mathbf{r}_j), \phi_j(\mathbf{r}_j^0), \rho_j(\mathbf{r}_j'')\}_{j=1}^k, \tau(\mathbf{r}_{\text{id}}^*)), \\ \mathbf{c}'_3 = \text{COM}(\{\mathcal{T}_{\pi_j, \varphi_j, \tau}(\mathbf{e}_j'' + \mathbf{r}'_j), \xi_j(\mathbf{r}^{(j)} + \mathbf{r}_j), \phi_j(\widehat{\mathbf{e}}_j + \mathbf{r}_j^0), \rho_j(\mathbf{e}_j''' + \mathbf{r}_j'')\}_{j=1}^k, \\ \quad \tau(\text{id}^* + \mathbf{r}_{\text{id}}^*)). \end{array} \right.$$

6. Send CMT to \mathcal{V}' .

Receiving a challenge $\text{CH} \in \{1, 2, 3\}$, \mathcal{S} replies as follows:

1. If $\text{CH} = 1$, \mathcal{S} outputs \perp and aborts.
2. If $\text{CH} = 2$, \mathcal{S} sends

$$\text{RSP} = (\{\pi_j, \varphi_j, \xi_j, \phi_j, \rho_j, \mathbf{e}_j'' + \mathbf{r}_j', \mathbf{r}'^{(j)} + \mathbf{r}_j, \widehat{\mathbf{e}}_j + \mathbf{r}_j^0, \mathbf{e}_j''' + \mathbf{r}_j''\}_{j=1}^k, \tau, \text{id}^* + \mathbf{r}_{\text{id}^*}).$$

3. If $\text{CH} = 3$, \mathcal{S} sends $\text{RSP} = (\{\pi_j, \varphi_j, \xi_j, \phi_j, \rho_j, \mathbf{r}_j', \mathbf{r}_j, \mathbf{r}_j^0, \mathbf{r}_j''\}_{j=1}^k, \tau, \mathbf{r}_{\text{id}^*})$.

o If $\widetilde{\text{CH}} = 2$. \mathcal{S} does as follows:

1. Sample several random vectors and permutations,

$$\left\{ \begin{array}{l} \mathbf{r}'_1, \dots, \mathbf{r}'_k \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+2)3m}; \mathbf{r}_1, \dots, \mathbf{r}_k \xleftarrow{\$} \mathbb{Z}_q^{3m}; \mathbf{r}_1^0, \dots, \mathbf{r}_k^0 \xleftarrow{\$} \mathbb{Z}_q^{3m}; \\ \mathbf{r}''_1, \dots, \mathbf{r}''_k \xleftarrow{\$} \mathbb{Z}_q^{3(n+m+\ell)}; \pi_1, \dots, \pi_k \xleftarrow{\$} \mathcal{S}_{3m}; \varphi_1, \dots, \varphi_k \xleftarrow{\$} \mathcal{S}_{3m}; \\ \xi_1, \dots, \xi_k \xleftarrow{\$} \mathcal{S}_{3m}; \rho_1, \dots, \rho_k \xleftarrow{\$} \mathcal{S}_{3(n+m+\ell)}; \phi_1, \dots, \phi_k \xleftarrow{\$} \mathcal{S}_{3m}; \\ \tau \xleftarrow{\$} \mathcal{S}_{2\ell}; \mathbf{r}_{\text{id}^*} \xleftarrow{\$} \mathbb{Z}_q^{2\ell}; \text{id}^* \xleftarrow{\$} \mathbf{B}_{2\ell}; \mathbf{e}_1'', \dots, \mathbf{e}_k'' \xleftarrow{\$} \text{SecExt}(\text{id}^*); \\ \widehat{\mathbf{e}}_1, \dots, \widehat{\mathbf{e}}_k \xleftarrow{\$} \mathbf{B}_{3m}; \mathbf{r}'^{(1)}, \dots, \mathbf{r}'^{(k)} \xleftarrow{\$} \mathbf{B}_{3m}; \mathbf{e}_1''', \dots, \mathbf{e}_k''' \xleftarrow{\$} \mathbf{B}_{3m}. \end{array} \right.$$

2. Compute $\text{CMT} = (\mathbf{c}_1', \mathbf{c}_2', \mathbf{c}_3')$ as in $\widetilde{\text{CH}} = 1$.

3. Send CMT to \mathcal{V}' .

Receiving a challenge $\text{CH} \in \{1, 2, 3\}$, \mathcal{S} replies as follows:

1. If $\text{CH} = 1$, \mathcal{S} sends

$$\text{RSP} = (\{\mathcal{T}_{\pi_j, \varphi_j, \tau}(\mathbf{e}_j''), \mathcal{T}_{\pi_j, \varphi_j, \tau}(\mathbf{r}_j'), \xi_j(\mathbf{r}'^{(j)}), \xi_j(\mathbf{r}_j), \phi_j(\widehat{\mathbf{e}}_j), \phi_j(\mathbf{r}_j^0), \rho_j(\mathbf{e}_j'''), \rho_j(\mathbf{r}_j'')\}_{j=1}^k, \tau(\text{id}^*), \tau(\mathbf{r}_{\text{id}^*})).$$

2. If $\text{CH} = 2$, \mathcal{S} outputs \perp and aborts.

3. If $\text{CH} = 3$, \mathcal{S} sends $\text{RSP} = (\{\pi_j, \varphi_j, \xi_j, \phi_j, \rho_j, \mathbf{r}_j', \mathbf{r}_j, \mathbf{r}_j^0, \mathbf{r}_j''\}_{j=1}^k, \tau, \mathbf{r}_{\text{id}^*})$.

o If $\widetilde{\text{CH}} = 3$. \mathcal{S} does as follows:

1. Sample several random vectors and permutations as in $\widetilde{\text{CH}} = 2$.

2. Compute $\text{CMT} = (\mathbf{c}_1', \mathbf{c}_2', \mathbf{c}_3')$, where

$$\left\{ \begin{array}{l} \mathbf{c}_1' = \text{COM}(\{\pi_j, \varphi_j, \xi_j, \phi_j, \rho_j\}_{j=1}^k, \tau, \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j (\mathbf{e}_j'' + \mathbf{r}_j')) - \mathbf{u}, \\ \quad \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{r}'^{(j)} + \mathbf{r}_j \\ \widehat{\mathbf{e}}_j + \mathbf{r}_j^0 \end{pmatrix}) - \mathbf{b}, \\ \quad \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j (\mathbf{e}_j''' + \mathbf{r}_j'')) + \mathbf{Q} \cdot (\text{id}^* + \mathbf{r}_{\text{id}^*}) - \mathbf{c}), \\ \mathbf{c}_2' = \text{COM}(\{\mathcal{T}_{\pi_j, \varphi_j, \tau}(\mathbf{r}_j'), \xi_j(\mathbf{r}_j), \phi_j(\mathbf{r}_j^0), \rho_j(\mathbf{r}_j'')\}_{j=1}^k, \tau(\mathbf{r}_{\text{id}^*})), \\ \mathbf{c}_3' = \text{COM}(\{\mathcal{T}_{\pi_j, \varphi_j, \tau}(\mathbf{e}_j'' + \mathbf{r}_j'), \xi_j(\mathbf{r}'^{(j)} + \mathbf{r}_j), \phi_j(\widehat{\mathbf{e}}_j + \mathbf{r}_j^0), \\ \quad \rho_j(\mathbf{e}_j''' + \mathbf{r}_j'')\}_{j=1}^k, \tau(\text{id}^* + \mathbf{r}_{\text{id}^*})). \end{array} \right.$$

3. Send CMT to \mathcal{V}' .

Receiving a challenge $\text{CH} \in \{1, 2, 3\}$, \mathcal{S} replies as follows:

1. If $\text{CH} = 1$, \mathcal{S} sends as in $(\widetilde{\text{CH}} = 2, \text{CH} = 1)$.
2. If $\text{CH} = 2$, \mathcal{S} sends as in $(\widetilde{\text{CH}} = 1, \text{CH} = 2)$.
3. If $\text{CH} = 3$, \mathcal{S} outputs \perp and aborts.

Based on the statistically hiding property of the commitment scheme COM , the three distributions of CMT , CH , RSP are statistically close to those in the real interaction, \mathcal{S} outputs \perp and aborts with probability negligibly close to $1/3$. Furthermore, once \mathcal{S} does not halt, then a valid transcript will be given and the distribution of the transcript is statistically close to that in the real interaction, therefore \mathcal{S} can impersonate an honest prover \mathcal{P} with probability negligibly close to $2/3$.

Argument of Knowledge

To prove that our new protocol is an argument of knowledge for the relation $\mathcal{R}(n, k, \ell, q, m, \beta)$ (as shown in Sect. 4), thus to show the proposed protocol has the special soundness property.

In the followings, we show that if there exists a prover \mathcal{P}' (maybe cheating) who can correctly respond to 3 challenges $\text{CH} \in \{1, 2, 3\}$ corresponding to the same commitment CMT with the public inputs $(\mathbf{A}_0, \mathbf{A}_1^1, \mathbf{A}_2^2, \mathbf{B}, \mathbf{P}, \mathbf{u}, \mathbf{b}, \mathbf{c})$, then there exists an extractor \mathcal{K} who produces $(\text{id} = \text{bin}(i) \in \{0, 1\}^\ell, \mathbf{r}, \mathbf{e}_0, \mathbf{e}_1 \in \mathbb{Z}^m, \mathbf{e}' = (\mathbf{e}'_1, \mathbf{e}'_2, \text{bin}(i) \otimes \mathbf{e}'_2) \in \text{Sec}_\beta(\text{id}), \mathbf{s} \in \mathbb{Z}^n, \mathbf{e}_2 \in \mathbb{Z}^\ell)$ such that

$$(\mathbf{A}_0, \mathbf{A}_1^1, \mathbf{A}_2^2, \mathbf{B}, \mathbf{P}, \mathbf{u}, \mathbf{b}, \mathbf{c}; \text{id} = \text{bin}(i), \mathbf{e}', \mathbf{r}, \mathbf{e}_0, \mathbf{s}, \mathbf{e}_1, \mathbf{e}_2) \in \mathcal{R}.$$

Indeed, based on 3 valid responses $\text{RSP}_1, \text{RSP}_2, \text{RSP}_3$ given by \mathcal{P}' , the extractor \mathcal{K} can extract the following information:

$$\left\{ \begin{array}{l} \mathbf{t}_{\text{id}} \in \mathbb{B}_{2\ell}, \forall j \in \{1, 2, \dots, k\}, \mathbf{v}'_j \in \text{SecExt}(\mathbf{t}_{\text{id}}), \mathbf{v}_j \in \mathbb{B}_{3m}, \\ \mathbf{c}_1 = \text{COM}(\{\hat{\pi}_j, \hat{\varphi}_j, \hat{\xi}_j, \hat{\phi}_j, \hat{\rho}_j\}_{j=1}^k, \hat{\tau}, \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{x}'_j) - \mathbf{u}, \\ \quad \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{x}_j \\ \mathbf{x}_j^0 \end{pmatrix}) - \mathbf{b}, \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{x}''_j) + \mathbf{Q} \cdot \mathbf{x}_{\text{id}} - \mathbf{c}), \\ = \text{COM}(\{\tilde{\pi}_j, \tilde{\varphi}_j, \tilde{\xi}_j, \tilde{\phi}_j, \tilde{\rho}_j\}_{j=1}^k, \tilde{\tau}, \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{h}'_j), \\ \quad \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{h}_j \\ \mathbf{h}_j^0 \end{pmatrix}), \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{h}''_j) + \mathbf{Q} \cdot \mathbf{h}_{\text{id}}), \\ \mathbf{c}_2 = \text{COM}(\{\mathbf{w}'_j, \mathbf{w}_j, \mathbf{w}_j^0, \mathbf{w}''_j\}_{j=1}^k, \mathbf{v}_{\text{id}}) \\ = \text{COM}(\{\mathcal{T}_{\tilde{\pi}_j, \tilde{\varphi}_j, \tilde{\tau}}(\mathbf{h}'_j), \tilde{\xi}_j(\mathbf{h}_j), \tilde{\phi}_j(\mathbf{h}_j^0), \tilde{\rho}_j(\mathbf{h}''_j)\}_{j=1}^k, \tilde{\tau}(\mathbf{h}_{\text{id}})), \\ \mathbf{c}_3 = \text{COM}(\{\mathbf{v}'_j + \mathbf{w}'_j, \mathbf{v}_j + \mathbf{w}_j, \mathbf{v}_j^0 + \mathbf{w}_j^0, \mathbf{v}^{(j)} + \mathbf{w}''_j\}_{j=1}^k, \mathbf{t}_{\text{id}} + \mathbf{v}_{\text{id}}), \\ = \text{COM}(\{\mathcal{T}_{\hat{\pi}_j, \hat{\varphi}_j, \hat{\tau}}(\mathbf{x}'_j), \hat{\xi}_j(\mathbf{x}_j), \hat{\phi}_j(\mathbf{x}_j^0), \hat{\rho}_j(\mathbf{x}''_j)\}_{j=1}^k, \hat{\tau}(\mathbf{x}_{\text{id}})). \end{array} \right.$$

Based on the computationally binding property of COM, \mathcal{K} deduces that:

$$\left\{ \begin{array}{l} \mathbf{t}_{\text{id}} \in \mathbf{B}_{2\ell}, \hat{\tau} = \tilde{\tau}, \forall j \in \{1, \dots, k\}, \hat{\xi}_j = \tilde{\xi}_j, \hat{\phi}_j = \tilde{\phi}_j, \hat{\pi}_j = \tilde{\pi}_j, \\ \hat{\varphi}_j = \tilde{\varphi}_j, \hat{\rho}_j = \tilde{\rho}_j; \mathbf{t}_{\text{id}} = \tilde{\tau}(\mathbf{h}_{\text{id}}), \mathbf{t}_{\text{id}} + \mathbf{v}_{\text{id}} = \hat{\tau}(\mathbf{x}_{\text{id}}); \\ \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{x}'_j) - \mathbf{u} = \mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{h}'_j); \\ \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{x}_j \\ \mathbf{x}_j^0 \end{pmatrix}) - \mathbf{b} = \mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{h}_j \\ \mathbf{h}_j^0 \end{pmatrix}); \\ \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{x}''_j) + \mathbf{Q} \cdot \mathbf{x}_{\text{id}} - \mathbf{c} = \mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{h}''_j) + \mathbf{Q} \cdot \mathbf{h}_{\text{id}}; \\ \mathbf{w}'_j = \mathcal{T}_{\tilde{\pi}_j, \tilde{\varphi}_j, \tilde{\tau}}(\mathbf{h}'_j), \mathbf{v}'_j + \mathbf{w}'_j = \mathcal{T}_{\tilde{\pi}_j, \tilde{\varphi}_j, \tilde{\tau}}(\mathbf{x}'_j), \mathbf{v}'_j \in \text{SecExt}(\mathbf{t}_{\text{id}}); \\ \mathbf{w}_j = \tilde{\xi}_j(\mathbf{h}_j), \mathbf{v}_j + \mathbf{w}_j = \tilde{\xi}_j(\mathbf{x}_j), \mathbf{v}_j \in \mathbf{B}_{3m}. \\ \mathbf{w}_j^0 = \tilde{\phi}_j(\mathbf{h}_j^0), \mathbf{v}_j^0 + \mathbf{w}_j^0 = \tilde{\phi}_j(\mathbf{x}_j^0), \mathbf{v}_j^0 \in \mathbf{B}_{3m}. \\ \mathbf{w}''_j = \tilde{\rho}_j(\mathbf{h}_j), \mathbf{v}^{(j)} + \mathbf{w}''_j = \tilde{\rho}_j(\mathbf{x}''_j), \mathbf{v}^{(j)} \in \mathbf{B}_{3(n+m+\ell)}. \end{array} \right.$$

For $j \in \{1, 2, \dots, k\}$, let $\mathbf{e}'_j = \mathbf{x}'_j - \mathbf{h}'_j = \mathcal{T}_{\tilde{\pi}_j, \tilde{\varphi}_j, \tilde{\tau}}^{-1}(\mathbf{v}'_j)$, $\mathbf{r}^{(j)} = \mathbf{x}_j - \mathbf{h}_j = \tilde{\xi}_j^{-1}(\mathbf{v}_j)$, $\mathbf{e}_j^0 = \mathbf{x}_j^0 - \mathbf{h}_j^0 = \tilde{\phi}_j^{-1}(\mathbf{v}_j^0)$, $\mathbf{e}^{(j)} = \mathbf{x}''_j - \mathbf{h}''_j = \tilde{\rho}_j^{-1}(\mathbf{v}^{(j)})$, $\text{id}^* = \mathbf{x}_{\text{id}} - \mathbf{h}_{\text{id}} = \tilde{\tau}^{-1}(\mathbf{t}_{\text{id}})$, we have $\mathbf{e}'_j \in \text{SecExt}(\tilde{\tau}^{-1}(\mathbf{t}_{\text{id}})) = \text{SecExt}(\text{id}^*)$, $\mathbf{r}^{(j)}, \mathbf{e}_j^0 \in \mathbf{B}_{3m}$, $\mathbf{e}^{(j)} \in \mathbf{B}_{3(n+m+\ell)}$. Furthermore, $\mathbf{A}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{e}'_j) = \mathbf{u} \bmod q$, $\mathbf{B}^* \cdot (\sum_{j=1}^k \beta_j \begin{pmatrix} \mathbf{r}^{(j)} \\ \mathbf{e}_j^0 \end{pmatrix}) = \mathbf{b} \bmod q$ and $\mathbf{P}^* \cdot (\sum_{j=1}^k \beta_j \mathbf{e}^{(j)}) + \mathbf{Q} \cdot \text{id}^* = \mathbf{c} \bmod q$.

The knowledge extractor \mathcal{K} produces $\text{id} = \text{bin}(i) \in \{0, 1\}^\ell$, $\mathbf{e}' \in \text{Sec}_\beta(\text{id})$, \mathbf{r} , \mathbf{e}_0 , $\mathbf{e}_1 \in \mathbb{Z}^m$, $\mathbf{s} \in \mathbb{Z}^n$ and $\mathbf{e}_2 \in \mathbb{Z}^\ell$ as follows:

1. Let $\text{id}^* = (d_1, d_2, \dots, d_\ell, d_{\ell+1}, \dots, d_{2\ell}) = \tilde{\tau}^{-1}(\mathbf{t}_{\text{id}})$, we obtain $\text{bin}(i) = \text{id} = (d_1, d_2, \dots, d_\ell)$ and the index $i = \mathbf{g}_\ell^\top \cdot \text{bin}(i)$ where $\mathbf{g}_\ell = (1, 2, \dots, 2^{\ell-1})$.
2. Let $\mathbf{e}^* = \sum_{j=1}^k \beta_j \mathbf{e}'_j \in \mathbb{Z}_q^{(2\ell+2)3m}$, thus $0 < \|\mathbf{e}^*\|_\infty \leq \sum_{j=1}^k \beta_j \|\mathbf{e}'_j\|_\infty \leq \beta$. Since $\mathbf{e}'_j \in \text{SecExt}(\text{id}^*)$, there exist $\mathbf{e}_1^*, \mathbf{e}_2^* \in \mathbb{Z}^{3m}$ such that $\|\mathbf{e}_1^*\|_\infty, \|\mathbf{e}_2^*\|_\infty \leq \beta$ and $\mathbf{e}^* = (\mathbf{e}_1^*, \mathbf{e}_2^*, d_1 \mathbf{e}_2^*, d_2 \mathbf{e}_2^*, \dots, d_{2\ell} \mathbf{e}_2^*)$. Let $\mathbf{e}' = (\mathbf{e}'_1, \mathbf{e}'_2, d_1 \mathbf{e}'_2, \dots, d_\ell \mathbf{e}'_2) = (\mathbf{e}'_1, \mathbf{e}'_2, \text{bin}(i) \otimes \mathbf{e}'_2)$, where $\mathbf{e}'_1, \mathbf{e}'_2$ are obtained from $\mathbf{e}_1^*, \mathbf{e}_2^*$ by removing the last $2m$ coordinates. Thus $\mathbf{e}' \in \text{Sec}_\beta(\text{id})$, and

$$[\mathbf{A}_0 | \mathbf{A}_1 | \mathbf{g}_\ell \otimes \mathbf{A}_2] \cdot (\mathbf{e}'_1, \mathbf{e}'_2, \text{bin}(i) \otimes \mathbf{e}'_2) = \mathbf{u} \bmod q.$$

3. Let $\hat{\mathbf{r}} = \sum_{j=1}^k \beta_j \mathbf{r}^{(j)} \in \mathbb{Z}^{3m}$, $\hat{\mathbf{e}}_0 = \sum_{j=1}^k \beta_j \mathbf{e}_j^0 \in \mathbb{Z}^{3m}$, thus,

$$0 < \|\hat{\mathbf{r}}\|_\infty \leq \sum_{j=1}^k \beta_j \|\mathbf{r}^{(j)}\|_\infty \leq \beta, \quad 0 < \|\hat{\mathbf{e}}_0\|_\infty \leq \sum_{j=1}^k \beta_j \|\mathbf{e}_j^0\|_\infty \leq \beta.$$

Let $\mathbf{r} \in \mathbb{Z}^m$ be a vector obtained from $\hat{\mathbf{r}}$ by removing the last $2m$ coordinates, $\mathbf{e}_0 \in \mathbb{Z}^m$ obtained from $\hat{\mathbf{e}}_0$ by removing the last $2m$ coordinates. So $\mathbf{r} \in \mathbb{Z}^m$, $0 < \|\mathbf{r}\|_\infty \leq \beta$, $\mathbf{e}_0 \in \mathbb{Z}^m$, $0 < \|\mathbf{e}_0\|_\infty \leq \beta$ and $\mathbf{b} = (\mathbf{B}^\top \mathbf{A}_0) \cdot \mathbf{r} + \mathbf{e} \bmod q$.

4. Let $\hat{\mathbf{e}} = \sum_{j=1}^k \beta_j \mathbf{e}^{(j)} \in \mathbb{Z}^{3(n+m+\ell)}$, so $0 < \|\hat{\mathbf{e}}\|_\infty \leq \sum_{j=1}^k \beta_j \|\mathbf{e}^{(j)}\|_\infty \leq \beta$, let $\mathbf{e} \in \mathbb{Z}^{n+m+\ell}$ be a vector obtained from $\hat{\mathbf{e}}$ by removing the last $2(n+m+\ell)$

coordinates. Parse $\mathbf{e} = \begin{pmatrix} \mathbf{s} \\ \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix}$ where $\mathbf{s} \in \mathbb{Z}^n$, $\mathbf{e}_1 \in \mathbb{Z}^m$, $\mathbf{e}_2 \in \mathbb{Z}^\ell$, so $\|\mathbf{e}\|_\infty \leq \beta$,

and $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) = \mathbf{P}\mathbf{e} + (\mathbf{0}^m, [q/2] \text{bin}(i)) \bmod q$.

Finally, the knowledge extractor \mathcal{K} outputs a tuple

$$(\text{id} = \text{bin}(i) \in \{0, 1\}^\ell, \mathbf{e}' \in \text{Sec}_\beta(\text{id}), \mathbf{r}, \mathbf{e}_0, \mathbf{e}_1 \in \mathbb{Z}^m, \mathbf{s} \in \mathbb{Z}^n, \mathbf{e}_2 \in \mathbb{Z}^\ell),$$

which is a valid witness for $\mathcal{R} = (n, k, \ell, q, m, \beta)$. This concludes the proof.

Acknowledgments. The authors would like to thank the anonymous reviewers of FCS 2019 for their helpful comments and this research is supported by the National Natural Science Foundation of China under Grant 61772477.

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: STOC, pp. 99–108. ACM (1996). <https://doi.org/10.1145/237814.237838>
2. Bellare, M., Micciancio, D., Warinschi, B.: Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_38
3. Bellare, M., Shi, H., Zhang, C.: Foundations of group signatures: the case of dynamic groups. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 136–153. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30574-3_11
4. Boneh, D., Shacham, H.: Group signatures with verifier-local revocation. In: CCS, pp. 168–177. ACM (2004). <https://doi.org/10.1145/1030083.1030106>
5. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J.: Foundations of fully dynamic group signatures. In: Manulis, M., Sadeghi, A.-R., Schneider, S. (eds.) ACNS 2016. LNCS, vol. 9696, pp. 117–136. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39555-5_7
6. Camenisch, J., Neven, G., Rückert, M.: Fully anonymous attribute tokens from lattices. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. LNCS, vol. 7485, pp. 57–75. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32928-9_4
7. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_27
8. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-46416-6_22
9. Gao, W., Hu, Y., Zhang, Y., Wang, B.: Lattice-based group signature with verifier-local revocation. J. Shanghai JiaoTong Univ. (Sci.) **22**(3), 313–321 (2017). <https://doi.org/10.1007/s12204-017-1837-1>
10. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoor for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206. ACM (2008). <https://doi.org/10.1145/1374376.1374407>
11. Gordon, S.D., Katz, J., Vaikuntanathan, V.: A group signature scheme from lattice assumptions. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 395–412. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_23
12. Kawachi, A., Tanaka, K., Xagawa, K.: Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 372–389. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89255-7_23

13. Kiayias, A., Yung, M.: Secure scalable group signature with dynamic joins and separable authorities. *Int. J. Secur. Netw.* **1**(1/2), 24–45 (2006). <https://doi.org/10.1504/ijsn.2006.010821>
14. Laguillaumie, F., Langlois, A., Libert, B., Stehlé, D.: Lattice-based group signatures with logarithmic signature size. In: Sako, K., Sarkar, P. (eds.) *ASIACRYPT 2013*. LNCS, vol. 8270, pp. 41–61. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42045-0_3
15. Langlois, A., Ling, S., Nguyen, K., Wang, H.: Lattice-based group signature scheme with verifier-local revocation. In: Krawczyk, H. (ed.) *PKC 2014*. LNCS, vol. 8383, pp. 345–361. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_20
16. Libert, B., Ling, S., Mouhartem, F., Nguyen, K., Wang, H.: Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In: Cheon, J.H., Takagi, T. (eds.) *ASIACRYPT 2016*. LNCS, vol. 10032, pp. 373–403. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_13
17. Libert, B., Ling, S., Nguyen, K., Wang, H.: Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In: Fischlin, M., Coron, J.-S. (eds.) *EUROCRYPT 2016*. LNCS, vol. 9666, pp. 1–31. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_1
18. Libert, B., Mouhartem, F., Nguyen, K.: A lattice-based group signature scheme with message-dependent opening. In: Manulis, M., Sadeghi, A.-R., Schneider, S. (eds.) *ACNS 2016*. LNCS, vol. 9696, pp. 137–155. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39555-5_8
19. Ling, S., Nguyen, K., Roux-Langlois, A., Wang, H.: A lattice-based group signature scheme with verifier-local revocation. *Theor. Comput. Sci.* **730**, 1–20 (2018). <https://doi.org/10.1016/j.tcs.2018.03.027>
20. Ling, S., Nguyen, K., Stehlé, D., Wang, H.: Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In: Kurosawa, K., Hanaoka, G. (eds.) *PKC 2013*. LNCS, vol. 7778, pp. 107–124. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36362-7_8
21. Ling, S., Nguyen, K., Wang, H.: Group signatures from lattices: simpler, tighter, shorter, ring-based. In: Katz, J. (ed.) *PKC 2015*. LNCS, vol. 9020, pp. 427–449. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_19
22. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Lattice-based group signatures: achieving full dynamism with ease. In: Gollmann, D., Miyaji, A., Kikuchi, H. (eds.) *ACNS 2017*. LNCS, vol. 10355, pp. 293–312. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61204-1_15
23. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Forward-secure group signatures from lattices. In: Ding, J., Steinwandt, R. (eds.) *PQCrypto 2019*. LNCS, vol. 11505, pp. 44–64. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25510-7_3
24. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Constant-size group signatures from lattices. In: Abdalla, M., Dahab, R. (eds.) *PKC 2018*. LNCS, vol. 10770, pp. 58–88. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76581-5_3
25. Micciancio, D., Peikert, C.: Hardness of SIS and LWE with small parameters. In: Canetti, R., Garay, J.A. (eds.) *CRYPTO 2013*. LNCS, vol. 8042, pp. 21–39. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40041-4_2
26. Nguyen, P.Q., Zhang, J., Zhang, Z.: Simpler efficient group signatures from lattices. In: Katz, J. (ed.) *PKC 2015*. LNCS, vol. 9020, pp. 401–426. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_18

27. Perera, M.N.S., Koshiha, T.: Fully dynamic group signature scheme with member registration and verifier-local revocation. In: Ghosh, D., Giri, D., Mohapatra, R., Sakurai, K., Savas, E., Som, T. (eds.) ICMC 2018. PROMS, vol. 253, pp. 399–415. Springer, Singapore (2018). https://doi.org/10.1007/978-981-13-2095-8_31
28. Perera, M.N.S., Koshiha, T.: Zero-knowledge proof for lattice-based group signature schemes with verifier-local revocation. In: Barolli, L., Kryvinska, N., Enokido, T., Takizawa, M. (eds.) NBI S 2018. LNDECT, vol. 22, pp. 287–302. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-98530-5_68
29. Perera, M.N.S., Koshiha, T.: Achieving strong security and verifier-local revocation for dynamic group signatures from lattice assumptions. In: Katsikas, S.K., Alcaraz, C. (eds.) STM 2018. LNCS, vol. 11091, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01141-3_1
30. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 145–166. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_8
31. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC, pp. 84–93. ACM (2005). <https://doi.org/10.1145/1060590.1060603>
32. Zhang, Y., Hu, Y., Gao, W., Jiang, M.: Simpler efficient group signature scheme with verifier-local revocation from lattices. KSII Trans. Internet Inf. Syst. **10**(1), 414–430 (2016). <https://doi.org/10.3837/tiis.2016.01.024>



Post-Quantum Pseudorandom Functions from Mersenne Primes

Jiehui Nan^(✉), Mengce Zheng, and Honggang Hu^(✉)

Key Laboratory of Electromagnetic Space Information,
Chinese Academy of Sciences, School of Information Science and Technology,
University of Science and Technology of China, Hefei 230027, China
{ustcnjh,mczheng}@mail.ustc.edu.cn, hghu2005@ustc.edu.cn

Abstract. Pseudorandom functions (PRFs) serve as a fundamental cryptographic primitive that is essential for encryption, identification and authentication. The concept of PRFs first formalized by Goldreich, Goldwasser, and Micali (JACM 1986), and their construction is based on length-doubling pseudorandom generators (PRGs) by using the tree-extension technique. Subsequently, Naor and Reingold proposed a construction based on synthesizers (JACM 2004) which can be instantiated from factoring and the Diffie-Hellman assumption. Recently, some efficient constructions were proposed in the post-quantum background. Banerjee, Peikert, and Rosen (Eurocrypt 2012) constructed relatively more efficient PRFs based on “learning with error” (LWE). Soon afterwards, Yu and Steinberger (Eurocrypt 2016) proposed two efficient constructions of randomized PRFs (with public coin as a parameter) from “learning parity with noise” (LPN). In this paper, we construct standard and randomized PRFs via Mersenne prime assumptions which were proposed by Aggarwal et al. (Crypto 2018) as new post-quantum candidate hardness assumptions. In contrast with Yu and Steinberger’s constructions, our first construction could have the same parameters to their second construction but not needs extra public coin and our second construction has a smaller public coin and key size comparing with their first construction.

Keywords: Mersenne prime problem · Pseudorandom functions · Pseudorandom generators

1 Introduction

Pseudorandom functions (PRFs) play a central role in symmetric cryptography, and the concept of PRFs was first rigorously defined by Goldreich, Goldwasser, and Micali [9]. Given a PRF family, most central goals of symmetric cryptography such as encryption, authentication, and identification have simple solutions that make efficient use of the PRF. Informally, a family of deterministic functions is pseudorandom if no efficient adversary, given adaptive oracle access

to a randomly chosen function from the family, can distinguish it from a uniform random function. The seminal GGM construction of PRFs [9] is based on any length-doubling pseudorandom generators (and hence on any one-way functions). Subsequently, using pseudorandom synthesizers as building blocks, Naor and Reingold [17] proposed a new generic construction of PRFs. Synthesizers, not as well understood as PRGs, in particular do not have many candidate instantiations and most known instantiations rely on assumptions of number theory.

Later, Naor and Reingold [19] gave direct constructions of PRFs from concrete number-theoretic assumptions such as decision Diffie-Hellman, RSA, and factoring. Banerjee, Peikert, and Rosen [3] constructed relatively more efficient PRFs based on the learning with error (LWE) assumption. More specifically, they observed that LWE for certain range of parameters implies a deterministic variant called learning with rounding (LWR), and that LWR in turn gives rise to pseudorandom synthesizers [17]. Despite that LWE is generalized from learning parity with noise (LPN), the derandomization technique used for LWE [3] does not seemingly apply to LPN, and thus it is an interesting open problem if low-depth PRFs can be based on (even a low-noise variant of) LPN. Yu and Steinberger [20] answered the above question and gave more efficient and parallelizable constructions of randomized PRFs from LPN under noise rate n^{-c} (for any constant $0 < c < 1$) and they can be implemented with a family of polynomial-size circuits.

Recently, Aggarwal et al. proposed the Mersenne Low Hamming Combination and Ratio Problem as new post-quantum candidate hardness assumptions to construct secure public-key encryption schemes [1] and we brief sketch these two problems when considering a Mersenne prime in the form $p = 2^n - 1$ (where n is prime). Informally speaking, Mersenne Low Hamming Combination Problem says that given a uniform random element R in \mathbb{Z}_p , $R \cdot A + B \pmod{p}$ is distinguishable from a uniform random element in \mathbb{Z}_p , where the secrets A and B are chosen uniformly at random from the elements in \mathbb{Z}_p with Hamming weight h . For the same distribution of A and B , Mersenne Low Hamming Ratio Problem says that $A \cdot B^{-1} \pmod{p}$ is indistinguishable from a uniform random element in \mathbb{Z}_p . Regarding the practical aspect, Mersenne prime problem provides efficiency due to its reliance on Mersenne primes and recently Ferradi and Xagawa [8] presented a novel and efficient secret-key authentication and MAC based on the Mersenne prime problem. We try to construct PRFs from the Mersenne prime problems in this paper and this work is inspired by the Yu and Steinberger's work [20]. Here, we briefly review the main idea of Yu and Steinberger's construction [20] based on a variant LPN problem which says that given a uniform random binary matrix R , it is hard to distinguish $(R \cdot s + e) \pmod{2}$ from a uniform random binary vector, where s and e are two binary vectors and every coordinate component of s and e is independently chosen from Bernoulli distribution. Setting the R as the public coin, they then extract and sample s and e from a weak random source using the public coin. The above operation directly gives a PRG, and then a PRF can be achieved via the GGM transformation. In fact, truly randomness is a scarce resource in practical application

and based on the Mersenne Low Hamming Ratio Problem, we achieve a PRF having same input/output length and key size to one of their constructions [20] but can remove the public coin (hence can save amount of truly randomness), which comes from an observation that the representation of Mersenne Low Hamming Ratio Problem $A \cdot B^{-1} \pmod{p}$ does not need the public coin setting in contrast with the LPN problem defined within a public matrix R . Our second construction of PRFs based on Mersenne Low Hamming Combination Problem can also reduce the public coin size for a PRF with same parameters of Yu and Steinberger's another construction [20] since the form of $R \cdot A + B \pmod{p}$ can be viewed as a ring variant of LPN in a sense.

In this paper, we construct PRFs based on the Mersenne prime problems, which provide post-quantum security promise. Since the structure of Mersenne Low Hamming Combination Problem [1] over Mersenne prime ring (also a field) can be viewed as the ring version of LPN and the Mersenne Low Hamming Ratio Problem [1] has a more succinct form, our two constructions can both save a large randomness used to construct PRFs in contrast with Yu and Steinberger's constructions [20] but our first construction does not have some efficient parallel algorithms. Concretely, our first construction of standard PRFs does not need the extra public coin, but their constructions need the public coin due to the randomized feature of LPN. Our second construction of randomized PRFs needs fewer public coin size than theirs due to the structure of Mersenne Low Hamming Combination Problem.

The rest of this paper is organized as follows. In Sect. 2, we introduce some definitions and notations. In Sect. 3, we present two new variants of Mersenne prime problem with a different distribution more efficiently to sample from. In Sect. 4, we construct standard PRFs from new variant of Mersenne Low Hamming Ratio Problem, and in Sect. 5, we construct randomized PRFs from new variant of Mersenne Low Hamming Combination Problem. Finally, we give conclusion in Sect. 6.

2 Preliminaries

2.1 Definitions and Notations

Throughout this paper, we denote the set $\{1, \dots, n\}$ by $[n]$. We use capital letters (e.g. X, Y) for random variables and distributions except the G and F , which are reserved for *PRGs* and *PRFs* respectively. We use standard letters (e.g. x, y) for values, and calligraphic letters (e.g. \mathcal{X}, \mathcal{E}) for sets and events. We denote the support of a random variable X by $\text{Supp}(X)$ referring the set of values on which X takes with non-zero probability, i.e., $\{x: \Pr[X = x] > 0\}$, and denote the cardinality of set \mathcal{S} by $|\mathcal{S}|$. We use χ_i^n , $i \leq n$, to denote a uniform distribution over set $\{e \in \{0, 1\}^n: \text{Ham}(e) = i\}$, where $\text{Ham}(e)$ denotes the Hamming weight of binary string e . The uniform distribution over $\{0, 1\}^n$ can be denoted by U_n . $X \sim D$ means that random variable X follows distribution D . $s \leftarrow S$ can be used to denote sampling an element s according to distribution S , and we use $s \stackrel{\$}{\leftarrow} S$ to denote sampling s uniformly from set S .

We also introduce some simplified notations in the following. Let λ to denote the security parameter, and n the instance parameter. Most other parameters are functions of n , and for the simplification, we often omit n when it is clear from the context. For example, $t = t(n) > 0$, $\epsilon = \epsilon(n) \in (0, 1)$, and we use $\text{poly}(n)$ denotes some polynomial function of n .

Aggarwal et al. recently introduced some new assumptions [1], called Mersenne prime assumptions, mimicking the NTRU over integers, relying on the properties of Mersenne prime in the ring of integers modulo p denoted by Z_p instead of polynomial ring $Z_q[x]/(x^n - 1)$. Before formally defining these Mersenne prime assumptions, we need to introduce some notations and definitions. When it is clear from the context, the arithmetic operations are all defined over Z_p throughout this paper.

Let $p = 2^k - 1$ for a positive integer k . We call p a Mersenne number if k is a prime, and if $2^k - 1$ is itself a prime number, then it is called a Mersenne prime. Note that if $k = k_1 k_2$ is a composite number for $k_1, k_2 > 1$, then $2^{k_1} - 1$ and $2^{k_2} - 1$ all divide p , and hence p is not a prime. For example, $2^2 - 1$, $2^3 - 1$, $2^5 - 1$, $2^{13} - 1$, and $2^{17} - 1$ are some smallest Mersenne primes, and until now, the largest Mersenne prime known is $2^{82,589,933} - 1$ which was found on December 21, 2018.

The Hamming weight of an n -bit binary string x is the total number of 1's in x and is denoted by $\text{Ham}(x)$. Let $\text{seq} : Z_p \rightarrow \{0, 1\}^n$ be the map that to $x \in Z_p$ associates the binary string $\text{seq}(x)$ representing x . The map $\text{int} : \{0, 1\}^n \rightarrow Z_p$ sends a binary string y into an integer modulo p represented by y and note that $\text{int}(1^n) = 0$. Clearly, seq and int are inverse functions between Z_p and $\{0, 1\}^n \setminus \{1^n\}$ in a canonical way. This bijection between Z_p and $\{0, 1\}^n \setminus \{1^n\}$ can be used to define *addition* and *multiplication* over $\{0, 1\}^n$ in the natural way: for $x, y \in \{0, 1\}^n$, let $x + y = \text{seq}(\text{int}(x) + \text{int}(y))$, and $x \cdot y = \text{seq}(\text{int}(x) \cdot \text{int}(y))$.

Definition 1 (Mersenne Low Hamming Ratio Assumption [1]). *Given an n -bit Mersenne prime $p = 2^n - 1$, and an integer h , the advantage of any probabilistic polynomial time (PPT) adversary running in time $\text{poly}(n)$ in attempting to distinguish between $\text{seq}(\text{int}(A)/\text{int}(B))$ and R is at most $\text{poly}(n)/2^\lambda$, where R is an n -bit uniform random string, and $A, B \sim \chi_h^n$.*

Definition 2 (Mersenne Low Hamming Combination Assumption [1]). *Given an n -bit Mersenne prime $p = 2^n - 1$, and an integer h , the advantage of any probabilistic polynomial time (PPT) adversary running in time $\text{poly}(n)$ in attempting to distinguish between*

$$(R_1, R_1 \cdot A + B) \quad , \quad (R_1, R_2)$$

is at most $\text{poly}(n)/2^\lambda$, where $R_1, R_2 \sim U_n$, and $A, B \sim \chi_h^n$.

In next section, we define two new variants of Mersenne prime problem based on the above two assumptions. The main change is that we sample $A, B \in \{0, 1\}^n$ from a new distribution instead of the distribution χ_h^n . Next, we introduce some tools for constructing some randomness objects.

Definition 3 (Pairwise independent hashing). A function family $\mathcal{H} = \{h_a : \{0, 1\}^n \rightarrow \{0, 1\}^m, a \in \{0, 1\}^l\}$ is **pairwise independent** if for any $x_1 \neq x_2 \in \{0, 1\}^n$ and any $v \in \{0, 1\}^{2m}$, it holds that

$$\Pr_{a \stackrel{\$}{\leftarrow} \{0,1\}^l} [(h_a(x_1), h_a(x_2)) = v] = 2^{-2m}$$

Definition 4 (Universal hashing). A function family $\mathcal{H} = \{h_a : \{0, 1\}^n \rightarrow \{0, 1\}^m, a \in \{0, 1\}^l\}$ is **universal** if for any $x_1 \neq x_2 \in \{0, 1\}^n$, it holds that

$$\Pr_{a \stackrel{\$}{\leftarrow} \{0,1\}^l} [(h_a(x_1) = h_a(x_2))] \leq 2^{-m}$$

CONCRETE CONSTRUCTIONS. There are a few constructions of pairwise independent hashing family. Here, we introduce a simple construction from linear algebra. In this construction, for any $m \leq n$, the length of description for the function family \mathcal{H} is $l = \theta(n)$, where any $h \in \mathcal{H}$ can be computed efficiently. We describe the construction in the following:

$$\mathcal{H}_1 = \{h_{a,b} : \{0, 1\}^n \rightarrow \{0, 1\}^m | h_{a,b}(x) \stackrel{def}{=} a \cdot x \oplus b, a \in \{0, 1\}^{n+m-1}, b \in \{0, 1\}^m\}$$

where $a \in \{0, 1\}^{n+m-1}$ is interpreted as an $n \times m$ Toeplitz matrix, ‘ \cdot ’ and ‘ \oplus ’ denote matrix-vector multiplication and addition over $GF(2)$ respectively.

It is easy to see that pairwise independent hashing means universal hashing. We can achieve a simple construction of universal hashing from the above construction, simply setting $b = \vec{0}$, where $\vec{0}$ is a m -dimension zero vector. We have:

$$\mathcal{H}_2 = \{h_a : \{0, 1\}^n \rightarrow \{0, 1\}^m | h_a(x) \stackrel{def}{=} a \cdot x, a \in \{0, 1\}^{n+m-1}\}$$

Finally, we introduce the concepts of entropy, statistical distance, and extractor. For a random variable X and any $x \in \text{Supp}(X)$, the sample-entropy of x with respect to X is defined as $\mathbf{H}_X(x) \stackrel{def}{=} \log(1/\Pr[X = x])$. Then we define the Shannon entropy and min-entropy respectively as follows:

$$\mathbf{H}_1(X) \stackrel{def}{=} \mathbb{E}_{x \leftarrow X}[\mathbf{H}_X(x)], \mathbf{H}_\infty(X) \stackrel{def}{=} \min_{x \in \text{Supp}(X)} \mathbf{H}_X(x)$$

A random variable X of length n is called an (n, k) -min-entropy source if $\mathbf{H}_\infty(X) \geq k$. Given two random variable X and Y , the statistical distance between them is denoted by

$$\Delta(X, Y) \stackrel{def}{=} \frac{1}{2} \sum_{x \in \text{Supp}(X) \cup \text{Supp}(Y)} |\Pr[X = x] - \Pr[Y = x]|$$

We use $\Delta(X, Y|Z)$ as a shorthand for $\Delta((X, Z), (Y, Z))$. For any $\epsilon > 0$, we say that two distribution X and Y are ϵ -close, denoted by $X \approx_\epsilon Y$, if $\Delta(X, Y) \leq \epsilon$. We can now define the randomness extractors. Informally, they are functions that transform a weak random source into an almost uniform distribution using a small number of additional uniform random bits and we denote the length of it by d .

Definition 5 (Randomness extractor). A function $\text{Ext} : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$ is a (k, ϵ) extractor if for any (n, k) -min-entropy source X independent of U_d , we have $(\text{Ext}(X, U_d), U_d) \approx_\epsilon (U_m, U_d)$, where U_m is independent of X and U_d .

In fact, any universal hashing function is a good extractor, and we show the result as a lemma in the following:

Lemma 1 (Leftover Hash Lemma (LHL) [13,14]). Fix $\epsilon > 0$. Let \mathcal{H} be a universal hash family of size 2^d with input length n and output length $m = k - 2\log(1/\epsilon)$, and $H \xleftarrow{\$} \mathcal{H}$. For any (n, k) -min-entropy source X independent of H , we have

$$\Delta(H(X), U_m | H) \leq \epsilon/2,$$

i.e. $\text{Ext}(X, H) = H(X)$ is a $(k, \epsilon/2)$ extractor.

Our new constructions of PRGs and PRFs include standard form and randomized form, and the later form introduced in [20] is the generalization of the first. Thanks to the structure of the Mersenne Low Hamming Ratio Assumption, we could construct the standard form without the public coin directly. We also construct the randomized form in respect of the randomized feature of the Mersenne Low Hamming Combination Assumption. However, our construction needs smaller size of public coin comparing with Yu and Steinberger’s construction [20] from LPN with the same input and output length. Next, we define the randomized PRGs and PRFs, and treat the standard form as the special case.

Definition 6 (Randomized PRGs on weak seeds). Let $k \leq l_1 < l_2, l_3, t, \epsilon$ be functions of parameter n . An efficient functions family ensembles $\mathcal{G} = \{G_a : \{0, 1\}^{l_1} \rightarrow \{0, 1\}^{l_2}, a \in \{0, 1\}^{l_3}\}_{n \in \mathbb{N}}$ is a (t, ϵ) randomized PRG if for every probabilistic distinguisher D of running time t and for any (l_1, k) -min-entropy source K as a seed, it holds that

$$\left| \Pr_{K \sim U_{l_1}, A \sim U_{l_3}} [D(G_A(K), A) = 1] - \Pr_{A \sim U_{l_3}} [D(U_{l_2}, A) = 1] \right| \leq \epsilon$$

The stretch factor of \mathcal{G} is l_2/l_1 . Standard PRGs are a special case for empty a and on uniform random seed.

Definition 7 (Randomized PRFs on weak keys). Let $k \leq l_1, l_2, l_3, l, t, \epsilon$ be functions of parameter n . An efficient functions family ensembles $\mathcal{F} = \{F_{k,a} : \{0, 1\}^l \rightarrow \{0, 1\}^{l_2}, k \in \{0, 1\}^{l_1}, a \in \{0, 1\}^{l_3}\}_{n \in \mathbb{N}}$ is a (q, t, ϵ) randomized PRF if for every oracle-aided probabilistic distinguisher D of running time t and bounded by q queries and for any (l_1, k) -min-entropy source K as a key, it holds that

$$\left| \Pr_{K \sim U_{l_1}, A \sim U_{l_3}} [D^{F_{K,A}}(A) = 1] - \Pr_{A \sim U_{l_3}} [D^R(A) = 1] \right| \leq \epsilon$$

where R denotes a random function distribution ensemble mapping from l bits to l_2 bits. Standard PRFs are a special case for empty a and on uniform random key.

3 New Variant Mersenne Prime Problems

In this section, we define two Mersenne prime problems associated with a new distribution efficiently to sample from, and we denote it by $\psi_{h/n}^n$ for integers h and n . We propose these problems for two reasons: the first one is that Mersenne Low Hamming Ratio and Combination Assumptions seem correct and no efficient attack has been found on them until now; the other one is that the $\psi_{h/n}^n$ is nearly close to a convex combination of $\chi_h^n, \chi_{h+1}^n, \dots, \chi_{2h}^n$ for $h = n^c$, where $0 < c < 1$ is a constant. This fact could ensure the Hamming weight of the new distribution to be not too small (i.e. its lower bound is h) and also not too large (i.e. its upper bound is $2h$). In the primary assumptions, the low Hamming weight could not be too small for the security and too large for the efficiency.

We define the distribution ψ_μ^m by the Algorithm 1 which was introduced by Yu and Steinberger in [20], where m is length of binary string sampled from the distribution and $0 < \mu < 1$ is Hamming weight ratio, and we denote the Algorithm 1 as a function by $\text{Sample}_\psi(\cdot)$.

Algorithm 1. Sampling distribution ψ_μ^m

Require:

$2\mu m \log m$ random bits (assume WLOG that m is a power of 2)

Ensure:

The distribution ψ_μ^m satisfies Lemma 2.

- 1: Sample random $z_1, \dots, z_{\mu m}$ of Hamming weight 1, i.e., for every $i \in [m]$ $z_i \xleftarrow{\$} \{z \in \{0, 1\}^m : |z| = 1\}$.
 {E.g., to sample z_1 with randomness $r_1 \dots, r_{\log m}$, simply let each $(z_1, \dots, z_{\log m})$ -th bit of z_1 to be $r_1^{b_1} \wedge \dots \wedge r_{\log m}^{b_{\log m}}$, where $r_j^{b_j} \stackrel{\text{def}}{=} r_j$ for $b_j = 0$ and $r_j^{b_j} \stackrel{\text{def}}{=} \neg r_j$ otherwise.}
 - 2: Output the bitwise-OR of the vectors $z_1, \dots, z_{\mu m}$.
-

Lemma 2 ([20]). *The distribution ψ_μ^m (via Algorithm 1) is $2^{-\Omega(\mu m \log(1/\mu))}$ -close to a convex combination of $\chi_{\mu m}^m, \chi_{\mu m+1}^m, \dots, \chi_{2\mu m}^m$.*

The proof of Lemma 2 can be referred to [20]. Note that as conditioned on $\text{Ham}(\psi_\mu^m) = \bar{h}$ (assume that $h \leq \bar{h} \leq 2h$), ψ_μ^m hits every $x \in \{0, 1\}^m$ of Hamming weight $\text{Ham}(x) = \bar{h}$ with equal probability. For some suitable parameters, we have a conclusion in the following from the above lemma directly.

Corollary 1. *Consider an instance parameter n , let $h = n^c$ for a constant $0 < c < 1$, $m = n$, and $\mu = h/m = n^{c-1}$, then the distribution $\psi_{h/n}^n$ is $2^{-\Omega(n^c/\log n)}$ -close to a convex combination of $\chi_h^n, \chi_{h+1}^n, \dots, \chi_{2h}^n$, i.e. the distribution $\psi_{h/n}^n$ is the convex combination of $\chi_h^n, \chi_{h+1}^n, \dots, \chi_{2h}^n$ with overwhelming probability.*

Now, we define the new variants of Mersenne prime problem in the following:

Definition 8 (X-Mersenne Low Hamming Ratio Problem(X-MLHRP)). Given an n -bit Mersenne prime $p = 2^n - 1$, and let t and ϵ all be function of n . Let X be a distribution having low Hamming weight with overwhelming probability. The X-Mersenne Low Hamming Ratio Problem is (t, ϵ) -hard if for every probabilistic distinguisher D running in time t , it holds that

$$|Pr[D(\text{seq}(\text{int}(A)/\text{int}(B)))] - Pr[D(R)]| \leq \epsilon$$

where $A, B \sim X$, and $R \sim U_n$.

Definition 9 (X-Mersenne Low Hamming Combination Problem (X-MLHCP)). Given an n -bit Mersenne prime $p = 2^n - 1$, and let t and ϵ all be function of n . Let X be a distribution having low Hamming weight with overwhelming probability. The X-Mersenne Low Hamming Combination Problem is (t, ϵ) -hard if for every probabilistic distinguisher D running in time t , it holds that

$$|Pr[D((R_1, R_1 \cdot A + B)))] - Pr[D((R_1, R_2)))]| \leq \epsilon$$

where $A, B \sim X$, and $R_1, R_2 \sim U_n$.

It is easy to see that χ_h^n -Mersenne prime problems for some low Hamming weight h come from the primary Mersenne prime assumptions introduced in the previous section. Our constructions of PRGs and PRFs are based on the $\psi_{n^c}^n$ -Mersenne Low Hamming Ratio (resp. Combination) Problem for standard (resp. randomized) form, where $0 < c < 1$ is a constant. Corollary 1 shows that $\psi_{n^c}^n$ is convex combination of $\chi_h^n, \chi_{h+1}^n, \dots, \chi_{2h}^n$ with overwhelming probability, where $h = n^c$. This means that when we choose $A, B \leftarrow \psi_{n^c}^n$, there exists $h \leq \bar{h}_1, \bar{h}_2 \leq 2h$ such that $A \sim \chi_{\bar{h}_1}^n$ and $B \sim \chi_{\bar{h}_2}^n$ with overwhelming probability. From this view, if for any $h \leq \bar{h} \leq 2h$, $\chi_{\bar{h}}^n$ -Mersenne prime problems are hard, then we can assume heuristically that $\psi_{n^c}^n$ -Mersenne prime problems are also hard.

Finally, we discuss the hardness of these two Mersenne prime problems. For the primary Mersenne Low Hamming Combination Problem (i.e. χ_h^n -Mersenne Low Hamming Combination Problem), de Boer et al. [5] presented a meet-in-the-middle attack to solve it. Their classical attack runs in time $O(\binom{n-1}{h-1}^{1/2})$ and the quantum version runs in $O(\binom{n-1}{h-1}^{1/3})$, and they corresponds to roughly $\frac{1}{4}h \log(n)$ and $\frac{1}{6}h \log(n)$ bits security, respectively. On the other hand, the authors of [4, 5] presented an LLL-based algorithm for solving χ_h^n -Mersenne Low Hamming Ratio/Combination Problem and the running time of this attack is $O(2^{2h})$ on classical computer and $O(2^h)$ on quantum machine. As claim in [1], attacks against Mersenne prime problems cannot exceed the 2^h , where h is the Hamming weight parameter. For constraint of the application based on the Mersenne prime problems in [1, 8], it needs to set $h = \Theta(n^{1/2})$ (i.e. $c = \frac{1}{2}$). However, in our construction, it does not need this setting and in contrast, we can set $h = n^c$ for a constant $0 < c < 1$ naturally. As the Hamming weight of $\psi_{n^c}^n$ is between h

and $2h$ with overwhelming probability, we can regard h as the security parameter. For our new variants of (t, ϵ) -hard Mersenne prime problems, we can assume $\epsilon = 2^{-O(n^c)}$ heuristically with $t = \text{poly}(n)$.

Inspired by Yu and Steinberger’s work [20], we present two constructions of PRFs following the notations and descriptions from [20] in next two sections.

4 Standard PRFs from New Variant Mersenne Low Hamming Ratio Problem

In this section, we construct PRFs from $\psi_{n^{c-1}}^n$ -Mersenne Low Hamming Ratio Problem ($\psi_{n^{c-1}}^n$ -MLHRP) for a small constant c . We first give the roadmap in the following:

$$\begin{aligned} \psi_{n^{c-1}}^n\text{-MLHRP} &\Rightarrow \frac{n^{1-c}}{4\log n}\text{-stretch PRG} \xrightarrow{GGM} (\text{small domain}) \text{ PRF} \\ &\xrightarrow{\text{generalized Levin's trick}} (\text{security-preserving domain-extended}) \text{ PRF} \end{aligned}$$

4.1 A Direct Construction of PRGs from MLHRP

We can construct PRGs directly from the $\psi_{n^{c-1}}^n$ -MLHRP by using short uniform random seed to sample two elements following distribution $\psi_{n^{c-1}}^n$. Then we use these two elements to compute an n -bit binary string which is distinguishable from an n -bit uniform random binary string.

Theorem 1 (PRGs from $\psi_{n^{c-1}}^n$ -Mersenne Low Hamming Ratio Problem). *Given an n -bit Mersenne prime $p = 2^n - 1$, let $0 < c < 1$ be a constant. For any $4n^c \log n$ -bit uniform random binary string $\omega = (\omega_1, \omega_2)$ where $|\omega_1| = |\omega_2|$, assume that $\psi_{n^{c-1}}^n$ -Mersenne Low Hamming Ratio Problem is (t, ϵ) -hard, then $G : \{0, 1\}^{4n^c \log n} \rightarrow \{0, 1\}^n$, with $A = \text{Sample}_\psi(\omega_1)$, $B = \text{Sample}_\psi(\omega_2)$, and $G(\omega) = \text{seq}(\text{int}(A)/\text{int}(B))$, is a $(t - \text{poly}(n), \epsilon)$ -PRG with stretch factor $n^{1-c}/4\log n$.*

Proof. The proof is directly from the definition of $\psi_{n^{c-1}}^n$ -MERS Low Hamming Ratio Problem.

After building the PRGs from $\psi_{n^{c-1}}^n$ -MLHRP, then we can use the GGM transformation to achieve the PRFs from the PRGs. Here, we state a variant transformation by using a balanced 2^v -ary tree instead of the binary tree in [9]. We show the concrete transformation in the following theorem.

Theorem 2 (PRFs from PRGs [9]). *Let n be a instance parameter, $t = t(n)$, and $\epsilon = \epsilon(n)$. Let $\mathcal{G} = \{G : \{0, 1\}^m \rightarrow \{0, 1\}^{2^v \cdot m}\}_{n \in \mathbb{N}}$ be a (t, ϵ) standard PRGs (with stretch factor 2^v), where $v = O(\log n)$. For any seed k , parse $G(k)$ as 2^v blocks of m -bit strings:*

$$G(k) \stackrel{\text{def}}{=} G^{0\dots 00}(k) \| G^{0\dots 01} \| \dots \| G^{1\dots 11}(k)$$

where $G^{i_1 \cdots i_v}(k)$ denotes the $(i_1 \cdots i_v)$ -th m -bit block of $G(k)$. Then, for any $d \leq \text{poly}(n)$ and $q = q(n)$, the function family ensemble $\mathcal{F} = \{F_k : \{0, 1\}^{dv} \rightarrow \{0, 1\}^{2^v \cdot m}, k \in \{0, 1\}^m\}_{n \in \mathbb{N}}$, where

$$F_k(x_1 \cdots x_{dv}) \stackrel{\text{def}}{=} G(G^{x_{(d-1)v+1} \cdots x_{dv}}(\cdots G^{x_1 \cdots x_v}(k) \cdots)) \quad (1)$$

is a $(q, t - q \cdot \text{poly}(n), \epsilon)$ standard PRF.

Lemma 3 (Levin's Trick [15]). For any $l \leq n \in \mathbb{N}$, let R_1 be a random function distribution over $\{0, 1\}^l \rightarrow \{0, 1\}^n$, let \mathcal{H} be a family of universal hash functions from n bits to l bits, and let $H_1 \stackrel{\$}{\leftarrow} \mathcal{H}$. Let $R_1 \circ H_1(x) \stackrel{\text{def}}{=} R_1(H_1(x))$ be a function distribution over $\{0, 1\}^n \rightarrow \{0, 1\}^n$. Then, for any $q \in \mathbb{N}$ and any oracle aided D bounded by q queries, it holds that

$$\left| \Pr_{R_1, H_1} [D^{R_1 \circ H_1} = 1] - \Pr_R [D^R = 1] \right| \leq \frac{q^2}{2^{l+1}}, \quad (2)$$

where R is a random function distribution from n -bits to n -bits.

Theorem 3 (A direct PRF with domain-extention). Given an n -bit Mersenne prime $p = 2^n - 1$, and let $0 < c < 1$ be a constant. Assume $\psi_{n^{c-1}}^n$ -Mersenne Low Hamming Ratio Problem is (t, ϵ) -hard. Then for any (efficiently computable) $d = \omega(1) \leq O(n)$, any q , and a $4n^c \log n$ -bits uniform random key k ,

$$F_k : \{0, 1\}^n \rightarrow \{0, 1\}^n$$

is a $(q, t - q \text{poly}(n), O(dq\epsilon) + q^2 n^{-d})$ standard PRF.

Proof. By Theorem 1, for an n -bit Mersenne prime $p = 2^n - 1$ we can build a standard $(t - \text{poly}(n), \epsilon)$ -PRG from the (t, ϵ) -hard $\psi_{n^{c-1}}^n$ -MLHRP, where $0 < c < 1$ is a constant. This PRG has a stretch factor $2^v = n^{1-c}/4 \log n$, where $v = O(\log n)$. We plug it into the GGM construction with tree depth $d' = d \log n / ((1-c) \log n - \log \log n - 2) = \Theta(d/1-c)$ to get a $(q, t - q \text{poly}(n), O(dq\epsilon))$ standard PRF with input length $d'v = d \log n$ and output length n as follows: $F_k : \{0, 1\}^{d \log n} \rightarrow \{0, 1\}^n$, where k is a $4n^c \log n$ -bits uniform random key. To extend the input length by using the Levin's trick, we need expand uniform random k (by evaluating F_k on a few fixed points) into a pseudorandom tuple (\bar{k}, \bar{h}_1) , where $\bar{k} \in \{0, 1\}^{4n^c \log n}$ and \bar{h}_1 describes a universal hash function from n -bits to $l = d \log n$ -bits. Then we define the domain-extended PRF $\bar{F}_k(x) = F_{\bar{k}} \circ \bar{h}_1(x)$. For any oracle-aided distinguish D running in time $t - q \text{poly}(n)$ and making q queries, denote with $\delta_D(F_1, F_2) \stackrel{\text{def}}{=} |Pr[D^{F_1} = 1] - Pr[D^{F_2} = 1]|$ the advantage of D in distinguishing between function oracles F_1 and F_2 . We have the triangle inequality in the following:

$$\begin{aligned} \delta_D(F_{\bar{K}} \circ \bar{H}_1, R) &\leq \delta_D(F_{\bar{K}} \circ \bar{H}_1, F_K \circ H_1) + \delta_D(F_K \circ H_1, R_1 \circ H_1) \\ &\quad + \delta_D(R_1 \circ H_1, R) \\ &\leq O(dq\epsilon) + q^2 n^{-d}, \end{aligned} \quad (3)$$

where advantage is upper bounded by three terms. The first term is the indistinguishability between truly random tuple (K, H_1) and pseudorandom tuple (\bar{K}, \bar{H}_1) derived from the pseudorandom function F_K ; the second between F_K and random function R_1 and the last term between $R_1 \circ H_1$ and random function R . The conclusion is correct via Theorems 1, 2, and Lemma 3.

4.2 Going Beyond the Birthday Barrier

Unfortunately, for small $d = \omega(1)$ the security of the above PRF does not go beyond super-polynomial due to a birthday attack for the term $q^2 n^{-d}$. Next, we use the generalized Levin’s trick to go beyond the birthday bound. The essential idea dates back to [2, 16]. We show the technique lemma below.

Lemma 4 (Generalized Levin’s Trick [2, 16]). *For any $\kappa, l \leq n \in \mathbb{N}$, let R_1, \dots, R_κ be independent random function distributions over $\{0, 1\}^l \rightarrow \{0, 1\}^n$, let \mathcal{H} be a family of universal hash functions from n bits to l bits, and let $H_1 \dots H_\kappa$ be independent function distributions all uniform over \mathcal{H} . Let $F_{\mathbf{R}, \mathbf{H}}$ be a function distribution (induced by $\mathbf{R}=(R_1, \dots, R_\kappa)$ and $\mathbf{H}=(H_1, \dots, H_\kappa)$) over $\{0, 1\}^n \rightarrow \{0, 1\}^n$ defined as*

$$F_{\mathbf{R}, \mathbf{H}} \stackrel{\text{def}}{=} \bigoplus_{i=1}^{\kappa} R_i(H_i(x)).$$

Then, for any $q \in \mathbb{N}$ and any oracle aided D bounded by q queries, we have

$$|Pr[D^{F_{\mathbf{R}, \mathbf{H}}} = 1] - Pr[D^R = 1]| \leq \frac{q^{\kappa+1}}{2^{\kappa l}}$$

where R is a random function distribution from n -bits to n -bits.

Finally, we get the security-preserving domain-extension construction in the following.

Theorem 4 (A security-preserving PRF). *Given an n -bit Mersenne prime $p = 2^n - 1$, and let $0 < c < 1$ be a constant. Assume $\psi_{n^{c-1}}^n$ -Mersenne Low Hamming Ratio Problem is (t, ϵ) -hard. Then for any (efficiently computable) $d = \omega(1) \leq O(n)$, any $q \leq n^{d/3}$, and a $4n^c \log n$ -bits uniform random key k , there exists a $(q, t - qpoly(n), O(dq\epsilon))$ standard PRF*

$$\hat{F}_k : \{0, 1\}^n \rightarrow \{0, 1\}^n$$

Proof. First, we can get a $(q, t - qpoly(n), \epsilon)$ standard PRF on $4n^c \log n$ -bits uniform random key with input length $l = d \log n$ and output length n . Then we define a new PRF via generalized Levin’s trick to achieve security-preserving domain extension. For an independent random key vector $\mathbf{k} = (k_1, \dots, k_\kappa) \in \{0, 1\}^{4\kappa n^c \log n}$ and a universal hashing vector $\mathbf{h} = (h_1, \dots, h_\kappa) \in \mathcal{H}^\kappa$, we define a PRF $F_{\mathbf{k}, \mathbf{h}} : \{0, 1\}^n \rightarrow \{0, 1\}^n$ as follows:

$$F_{\mathbf{k}, \mathbf{h}}(x) \stackrel{\text{def}}{=} \bigoplus_{i=1}^{\kappa} F_{k_i}(h_i(x))$$

Let $\delta_D(F_1, F_2) \stackrel{\text{def}}{=} |Pr[D^{F_1} = 1] - Pr[D^{F_2} = 1]|$ be same in the proof of Theorem 3. We have that for any oracle-aided distinguisher D running in time $t - q\text{poly}(n)$ with $q \leq n^{d/3}$ bounded queries, a triangle inequality holds as follows:

$$\begin{aligned} \delta_D(F'_{K,H}, R) &\leq \delta_D(F'_{K,H}, F_{R,H}) + \delta_D(F_{R,H}, R) \\ &\leq O(\kappa dq\epsilon) + n^{d(1-2\kappa)/3} \\ &= O(\kappa dq\epsilon) + 2^{-\omega(n^c)} = O(\kappa dq\epsilon) \end{aligned} \tag{4}$$

where $\mathbf{K} = (K_1, \dots, K_\kappa)$ and $\mathbf{H} = (H_1, \dots, H_\kappa)$ are truly independent random string vectors, $F_{R,H} = \bigoplus_{i=1}^\kappa R_i(H_i(x))$, and $\mathbf{R} = (R_1, \dots, R_\kappa)$ is an independent random function distribution vector over $\{0, 1\}^l \rightarrow \{0, 1\}^n$. In fact, the first term of the second inequality is from hybrid argument (i.e. replacing every F_{K_i} with R_i one at a time), the second term of the second inequality follows from Lemma 4 with $l = d\log n$ and $q \leq n^{d/3}$, and the equalities follow by setting $\kappa = n^c$ to make the first term dominant.

The last problem is to generate multiple keys and universal hashing functions by using the single *PRF* key k multi-times for different inputs, i.e. we define $\hat{F}_k(x) \stackrel{\text{def}}{=} F'_{\mathbf{k},\mathbf{h}}(x)$, where $(\mathbf{k}, \mathbf{h}) = F_k(1) \| F_k(2) \| \dots \| F_k(O(\kappa))$. Using the hybrid technique again, we have $\delta_D(\hat{F}_K, F'_{K,H}) \leq O(\kappa dq\epsilon)$. Combining with these results, we finish the proof of Theorem 4.

Consider the same parameters in Theorem 4, let us review the Yu and Steinberger’s second construction [20]. To construct a PRF with input and output length n on a $\Theta(n^c \log n)$ -bits uniform key, they need a $O(n^2)$ size public coin. However, this extra randomness can be reduced in our construction.

5 Randomized PRFs with Short Public Coins

In this section, we apply the $\psi_{n^{c-1}}^n$ -Mersenne Low Hamming Combination Problem ($\psi_{n^{c-1}}^n$ -MLHCP) to built randomized PRFs similarly to construction from decisional ψ_{μ}^{n+q} -LPN $_{\mu,n}$ in [20], but we use a shorter public coin to extract an almost uniform distribution from a weak randomness source. The tools we used to construct randomized PRFs have mentioned in previous section.

Theorem 5 (A randomized PRGs from $\psi_{n^{c-1}}^n$ -Mersenne Low Hamming Combination Problem). *Given an n -bit Mersenne prime $p = 2^n - 1$. Let $0 < c < 1$ and $c < \delta < 1$. Assume $\psi_{n^{c-1}}^n$ -Mersenne Low Hamming Combination Problem is (t, ϵ) -hard. Then there exists a $(t - \text{poly}(n), O(\epsilon))$ -randomized PRG on $(n^\delta, O(n^c \log n))$ -weak seed (i.e. $(n^\delta, O(n^c \log n))$ -min-entropy source) with public coin size n and stretch factor $n^{1-\delta}$.*

Proof. Given an n -bit uniform random public coin $R \in \{0, 1\}^n$, we apply it as a universal hashing function H_R with input length n^δ and output length $4n^c \log n$ on the $(n^\delta, O(n^c \log n))$ -weak seed ω , and then we can achieve a distribution

that is $2^{-\Omega(n^c \log n)}$ -close to uniform random distribution $U_{4n^c \log n}$ via the leftover hash lemma. Set $H_R(K) = (S_1, S_2)$. We choose the secrets of $\psi_{n^{c-1}}^n$ -MLHCP with $\bar{A} = \text{Sample}_\psi(S_1)$ and $\bar{B} = \text{Sample}_\psi(S_2)$, and then define the randomized PRG as $G_R(\omega) = \text{seq}(\text{int}(R) \cdot \bar{A} + \bar{B})$. For any distinguisher D running in time $t - \text{poly}(n)$ against two different distributions X and Y , we denote with $\delta_D(X, Y) \stackrel{\text{def}}{=} |Pr[D(X) = 1] - Pr[D(Y) = 1]|$ the advantage of D in distinguishing between X and Y . For $R \sim U_n$, $A, B \sim \psi_{n^{c-1}}^n$, and a $(n^\delta, O(n^c \log n))$ -weak seed ω , we have the triangle inequality:

$$\begin{aligned} \delta_D(G_R(\omega), U_n) &\leq \delta_D(G_R(\omega), \text{seq}(\text{int}(R) \cdot A + B)) \\ &\quad + \delta_D(\text{seq}(\text{int}(R) \cdot A + B), U_n) \\ &\leq 2^{-\Omega(n^c \log n)} + \epsilon = O(\epsilon). \end{aligned} \tag{5}$$

The last equation is correct when we assume that $2^{-\Omega(n^c \log n)} \leq \epsilon$ for state of the art of attacks. It is easy to see that the stretch factor is $n^{1-\delta}$.

Theorem 6 (A randomized PRFs from $\psi_{n^{c-1}}^n$ -Mersenne Low Hamming Combination Problem). *Given an n -bit Mersenne prime $p = 2^n - 1$. Let $0 < c < 1$ and $c < \delta < 1$. Assume $\psi_{n^{c-1}}^n$ -Mersenne Low Hamming Combination Problem is (t, ϵ) -hard. Then for any (efficiently computable) $d = \omega(1) \leq O(n)$, any $q \leq n^{d/3}$, there exists a $(q, t - q\text{poly}(n), O(dq\epsilon))$ -randomized PRF on $(n^\delta, O(n^c \log n))$ -weak key (i.e. $(n^\delta, O(n^c \log n))$ -min-entropy source) with public coin size n .*

Proof. The Theorem 5 shows that we can apply an n -bits public coin R as an extractor to transform a weak seed into an almost uniform distribution, and then invoke the sample algorithm Sample_ψ on this distribution to produce two secrets of (t, ϵ) -hard $\psi_{n^{c-1}}^n$ -MLHCP, which implies a $n^{1-\delta}$ -stretch randomized PRG with input length n^δ and output length n . Using $2^v = \Theta(n^{1-\delta})$ -ary GGM transformation for $\Theta(d)$ times, we can obtain a small domain PRF and then achieve a security-preserving domain-extended PRF via the generalized Levin's trick. Finally, we can achieve a $(q, t - q\text{poly}(n), O(dq\epsilon))$ -randomized PRF on $(n^\delta, O(n^c \log n))$ -weak key by choosing some suitable parameters similar to the construction of standard PRFs in previous section.

By setting $\delta = (c + 1)/2 > c$, for an instance parameter n we can achieve a randomized PRF with input/output length n and on $(n^{(c+1)/2}, O(n^c \log n))$ -weak key only requiring an extra public coin with size n . However, in the same setting of the input/output length and weak key, Yu and Steinberger's first construction [20] needs a $O(n^2)$ size public coin. If we let δ be larger than c but smaller than $(c + 1)/2$, we can also achieve a randomized PRF with a smaller key size than theirs.

In fact, there only exists some finite instance parameters for our construction of PRFs. However, in practical usage, we can choose a large parameter and then use truncation technique to reduce the output length. We can also use the generalized Levin's trick to adjust the domain size.

6 Conclusion

In this paper, we construct standard and randomized PRFs based on Mersenne prime. We propose two new variants of Mersenne prime problems with a different distribution in contrast with the primary form, and the new distribution we used has an efficient sampling algorithm. Our work is inspired by Yu Yu and John Steinberger's work, and our construction is not efficient than theirs in a parallel way, but we can save more randomness than theirs with same parameters. Concretely, with same input/output length and key size, our first construction is for standard PRFs (i.e. without the public coin) but their corresponding construction needs public coin with size of square of input length. For the same input/output length and weak key source, the public coin size of their another construction has a linear factor than the public coin size of our second construction we needed, and further more, if we choose some suitable parameter, we can reduce the key size simultaneously.

Acknowledgment. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was partially supported by the National Natural Science Foundation of China (Grant No. 61632013).

References

1. Aggarwal, D., Joux, A., Prakash, A., Santha, M.: A new public-key cryptosystem via mersenne numbers. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10993, pp. 459–482. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96878-0_16
2. Bellare, M., Goldreich, O., Krawczyk, H.: Stateless evaluation of pseudorandom functions: security beyond the birthday barrier. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 270–287. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_17
3. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 719–737. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_42
4. Beunardeau, M., Connolly, A., Géraud, R., Naccache, D.: On the Hardness of the Mersenne Low Hamming Ratio Assumption. Cryptology ePrint Archive: Report 2017/522 (2017)
5. de Boer, K., Ducas, L., Jeffery, S., de Wolf, R.: Attacks on the AJPS mersenne-based cryptosystem. In: Lange, T., Steinwandt, R. (eds.) PQCrypto 2018. LNCS, vol. 10786, pp. 101–120. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-79063-3_5
6. Carter, J., Wegman, N.: Universal classes of hash functions. J. Comput. Syst. Sci. **18**(2), 143–154 (1979)
7. Döttling, N., Schröder, D.: Efficient pseudorandom functions via on-the-fly adaptation. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 329–350. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_16

8. Ferradi, H., Xagawa, K.: Post-Quantum Provably-Secure Authentication and MAC from Mersenne Primes. Cryptology ePrint Archive: Report 2019/409 (2019)
9. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. *J. ACM* **33**(4), 792–807 (1986)
10. Hoeffding, W.: Probability inequalities for sums of bounded random variables. *J. Am. Stat. Assoc.* **58**(301), 13–30 (1963)
11. Håstad, J., Impagliazzo, R., Levin, L., Luby, M.: Construction of pseudorandom generator from any one-way function. *SIAM J. Comput.* **28**(4), 1364–1396 (1999)
12. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring-based public key cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0054868>
13. Impagliazzo, R., Zuckerman, D.: To recycle random bits. In: 30th Annual Symposium on Foundations of Computer Science (FOCS 1989), pp. 12–24. IEEE, Research Triangle Park (1989)
14. Impagliazzo, R., Levin, L. A., Luby, M.: Pseudo-random generation from one-way functions. In: 21th Annual ACM Symposium on Theory of Computing (STOC 1989), pp. 12–24. ACM, Seattle (1989)
15. Levin, L.A.: One way functions and pseudorandom generators. *Combinatorica* **7**(4), 357–363 (1987)
16. Maurer, U.: Indistinguishability of random systems. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 110–132. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_8
17. Naor, M., Reingold, O.: Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.* **58**(2), 336–375 (1999)
18. Naor, M., Reingold, O., Rosen, A.: Pseudorandom Functions and Factoring. *SIAM J. Comput.* **31**(5), 1383–1404 (2002)
19. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. *J. ACM* **51**(2), 231–262 (2004)
20. Yu, Y., Steinberger, J.: Pseudorandom functions in almost constant depth from low-noise LPN. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9666, pp. 154–183. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_6

Signature



An Efficient Proxy Re-Signature Over Lattices

Mingming Jiang¹, Jinqiu Hou¹, Yuyan Guo^{1(✉)}, Yan Wang²,
and Shimin Wei¹

¹ School of Computer Science and Technology,
Huaibei Normal University, Huaibei 235000, China
guoyuyan428@163.com

² School of Mathematics Science, Huaibei Normal University,
Huaibei 235000, China

Abstract. In 2008, Libert and Vergnaud constructed the first multi-use unidirectional proxy re-signature scheme. In this scheme, the proxy can translate the signatures several times but only in one direction. Thus, two problems remain open. That is, to construct a multi-use unidirectional proxy re-signature scheme based on classical hardness assumptions, and to design a multi-use unidirectional proxy re-signature scheme with the size of signatures and the verification cost growing sub-linearly with the number of translations. This paper solves the first problem and sharply reduces the verification costs. We use the preimage sampleable algorithm to develop a multi-use unidirectional proxy re-signature scheme based on lattices, namely, the hardness of the Small Integer Solution (SIS) problem. The verification cost does not grow with the number of translations and the size of signatures grows linearly with the number of translations in this scheme. Furthermore, the proposal is secure in quantum environment.

Keywords: Lattice cryptography · Proxy re-signature scheme · Small Integer Solution (SIS) problem · Gaussian Sample · Multi-use

1 Introduction

Proxy re-signature is proposed by Blaze, Bleumer, and Strauss [1]. In a proxy re-signature scheme, a semi-trusted proxy is given some information that allows it to transform Alice's signature into Bob's signature on the same message, but the proxy cannot generate signatures for Alice or Bob on its own. In [1], the first proxy re-signature scheme is constructed and is proven to be multi-use and bidirectional. However, the proxy re-signature primitive was seldom noticed until 2005. In 2005, Ateniese and Hohenberger [2] formalized the definition of security and illustrated the applications of proxy re-signature schemes. What follows presents some properties that will be taken into account in a proxy re-signature scheme.

1. Unidirectional: the proxy only can turn the Alice's signatures into the Bob's signatures, but the reverse is not true.
2. Multi-use: a signature can be re-signed many times;
3. Private Proxy: re-signature keys are kept secret;
4. Transparent: we can not distinguish the re-signatures from the original signatures;

5. Key optimal: a user is only required to store a constant amount of secret data;
6. Non-interactive: the delegatee does not participate in the process of the generation of the proxy re-signature key;
7. Non-transitive: the re-signing rights cannot be re-delegated by the proxy;
8. Unlinkable: a re-signature cannot be linked to the one from which it was generated.

In [2], three proxy re-signature schemes were proposed: the first one is multi-use and bidirectional with a private re-signature key; the second one is single-use and unidirectional with a public re-signature key; the third one is single-use and unidirectional with a private re-signature key. The possible applications of a re-signature scheme may include the space-efficient proof, group signatures management, simplification of certificate management. However, it remains an open problem to design a multi-use unidirectional re-signature scheme. To solve this problem, Labert and Vergnaud [3] proposed two multi-use and unidirectional schemes with a private re-signature key based on the *l*-FlexDH assumption (in the random oracle model and the standard model, respectively). However, we are confronted with two open problems: one is to construct a multi-use unidirectional proxy re-signature scheme under the standard hardness assumptions; the other is to reduce the size of signatures and the verification costs. Sunitha and Amberker [4] proposed another multi-use unidirectional proxy re-signature scheme, but the scheme only obtains a forward security, and hence is not provably secure. Sunitha [5] constructed a proxy signature schemes that translates Alice's Schnorr/ElGamal/RSA signature to Bob's RSA signature, but failed to prove the security. Shao et al. [6] proposed the first multi-use bidirectional proxy re-signature scheme in the standard model and extended it to the ID-based case. Shao et al. [7] proposed the first unidirectional identity based proxy re-signature in the random oracle based on the Schnorr's signature and the Libert-Vergnaud proxy re-signature. Shao et al. [8] analyzed and improved the previous security model [2] and gave a unidirectional proxy re-signature scheme to meet the new security model. Yang et al. [9] first defined the security model for threshold proxy re-signature scheme, and then proposed two threshold proxy re-signature schemes based on the Ateniese-Hohenberger's and the Shao-Cao-Wang-Liang's approach. However, the four proposals were built from the intractability assumptions for factoring large integers or solving discrete logarithms. Thus, they are not secure in the quantum setting and hence it is meaningful to construct a proxy re-signature scheme secure in the quantum setting.

As an important class of post-quantum cryptography, lattice cryptography attracts more and more attentions in the cryptographic literature in recent years due to the elegant cryptographic properties. First, lattice cryptography only involves some linear operations on small integers, and hence results in an asymptotically low computational complexity. Second, the security is supported by the worst-case to average-case equivalence connections. Since the first proposals of a provably secure lattice signature scheme and a lattice IBE scheme due to Gentry et al. [10], we are witnessing a rapid development of lattice cryptography. Many lattice schemes are constructed, such as the lattice-based public key encryption schemes [11–14], identity-based encryption schemes [10, 15–17], fully homomorphic encryption [18–21] and lattice-based signatures schemes [10, 22] and signature schemes with particular properties [23–25].

1.1 Contributions

We aim at the open problems left by Libert and Vergnaud over lattices. In our scheme, the proxy re-signature key is generated by the Gaussian Sample algorithm. First, given two public keys $pk_1 = \mathbf{A}_1, pk_2 = \mathbf{A}_2$ of users 1 and 2 and the secret key of user 2, use the Gaussian Sample algorithm to generate the proxy re-signature key $\mathbf{S}_{1 \rightarrow 2}$, such that $\mathbf{A}_2 \mathbf{S}_{1 \rightarrow 2} = \mathbf{A}_1 \bmod q$. Second, gives an original signature \mathbf{e}_1 of user 1, and the re-signature $\mathbf{e}_2 = \mathbf{S}_{1 \rightarrow 2} \mathbf{e}_1$. We know that the proxy re-signature key $\mathbf{S}_{1 \rightarrow 2}$ has two properties: (1) its norm is small; (2) its distribution is statistically close to a Gaussian distribution. Then the distribution of the re-signature is statistically close to a Gaussian distribution and its norm is also small. Thus, the proxy re-signature has the same properties as the original signature.

1.2 Organization

In Sect. 2, we formalize the related notations, review the definitions of lattice and Gaussian distribution, introduce the lattice basis delegation technique, and define the Small Integer Solution hardness assumption on which the security of our scheme is based. We describe the definition and security model of a Proxy Re-Signature scheme in Sect. 3. In Sect. 4, we propose a Multi-Use Unidirectional Proxy Re-Signature scheme based on lattice in the random model. The scheme in the standard model is constructed in Sect. 5. Finally, the conclusion is given in Sect. 6.

2 Preliminaries

2.1 Notation

We denote sets of real numbers by \mathbb{R} and the integers by \mathbb{Z} , respectively. Vectors are written as bold italic lower-case letters, e.g. \mathbf{x} . The i -th component of \mathbf{x} is denoted by x_i . Matrices are written as bold italic capital letters, e.g. \mathbf{X} , and the i -th column vector of a matrix \mathbf{X} is denoted \mathbf{x}_i . The Euclidean norm l_2 norm of a vector x is denoted as

$\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \sqrt{\sum_{i=1}^n x_i^2}$. Generally, we abbreviate $\|\mathbf{x}\|_2$ as $\|\mathbf{x}\|$. The length of a matrix is defined as the norm of the longest column, namely, $\|\mathbf{X}\| = \max_i \|\mathbf{x}_i\|$, for $1 \leq i \leq k$.

2.2 Lattice

Let $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_m\} \in \mathbb{R}^{m \times m}$ be an $m \times m$ matrix whose columns are linearly independent vectors $\mathbf{b}_1, \dots, \mathbf{b}_m \in \mathbb{R}^m$. The m -dimensional lattice Λ generated by \mathbf{B} ,

$$\Lambda = \mathcal{L}(\mathbf{B}) = \left\{ \mathbf{y} \in \mathbb{R}^m \text{ s.t. } \exists \mathbf{x} \in \mathbb{Z}^m, \mathbf{y} = \mathbf{B}\mathbf{x} = \sum_{i=1}^m x_i \mathbf{b}_i \right\} \quad (1)$$

Here, we focus on inter lattices, i.e., \mathcal{L} is contained in \mathbb{Z}^m .

Definition 1. For q prime, $A \in \mathbb{Z}_q^{n \times m}$, $\mathbf{u} \in \mathbb{Z}_q^n$, define:

$$\Lambda_q^\perp(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \mathbf{A}\mathbf{e} = \mathbf{0}(\text{mod } q)\} \quad (2)$$

$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) := \{\mathbf{e} \in \mathbb{Z}^m \text{ s.t. } \mathbf{A}\mathbf{e} = \mathbf{u}(\text{mod } q)\} \quad (3)$$

Observe that if $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$, then $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \Lambda_q^\perp(\mathbf{A}) + \mathbf{t}$, hence $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ is a shift of $\Lambda_q^\perp(\mathbf{A})$.

Lemma 1 [26]. Let $q \geq 3$ be odd and $m = \lceil 6n \log q \rceil$. There is a probabilistic polynomial-time algorithm $\text{TrapGen}(q, n)$ that outputs two matrixes $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{T} \in \mathbb{Z}_q^{m \times m}$ such that \mathbf{A} is statistically close to a uniform matrix in $\mathbb{Z}_q^{n \times m}$ and \mathbf{T} is a basis for $\Lambda_q^\perp(\mathbf{A})$ satisfying

$$\|\tilde{\mathbf{T}}\| \leq O(\sqrt{n \log q}) \text{ and } \|\mathbf{T}\| \leq O(n \log q) \text{ with all but negligible probability in } n.$$

2.3 Discrete Gaussians

We briefly recall Discrete Gaussian Distributions over lattices.

For any positive parameter $\sigma > 0$ define the Gaussian function on \mathbb{R}^m centered at \mathbf{c} :

$$\forall \mathbf{x} \in \mathbb{R}^m, \rho_{\sigma, \mathbf{c}}(\mathbf{x}) = \exp\left(-\pi \|\mathbf{x} - \mathbf{c}\|^2 / \sigma^2\right) \quad (4)$$

For any $\mathbf{c} \in \mathbb{R}^m$, real $\sigma > 0$, and an m -dimensional Λ , define the Discrete Gaussian Distribution over Λ as:

$$\forall \mathbf{x} \in \mathbb{R}^m, D_{\Lambda, \sigma, \mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma, \mathbf{c}}(\mathbf{x})}{\rho_{\sigma, \mathbf{c}}(\Lambda)} = \frac{\rho_{\sigma, \mathbf{c}}(\mathbf{x})}{\sum_{\mathbf{x} \in \Lambda} \rho_{\sigma, \mathbf{c}}(\mathbf{x})} \quad (5)$$

Lemma 2 [10]. Let $q \geq 2$ and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $m > n$. Let \mathbf{T}_A be a basis for $\Lambda_q^\perp(\mathbf{A})$, $\sigma \geq \|\tilde{\mathbf{T}}_A\| \cdot \omega(\sqrt{\log m})$. Then for $\mathbf{c} \in \mathbb{R}^m$, $\mathbf{u} \in \mathbb{Z}_q^n$:

1. $\Pr[\mathbf{x} \sim D_{\Lambda_q^\perp(\mathbf{A}), \sigma} : \|\mathbf{x}\| > \sigma\sqrt{m}] \leq \text{negl}(n)$.
2. There is a polynomial-time algorithm $\text{SampleGaussian}(\mathbf{A}, \mathbf{T}_A, \sigma, \mathbf{c})$ that returns $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A})$ drawn from a distribution statistically close to $D_{\Lambda_q^\perp(\mathbf{A}), \sigma, \mathbf{c}}$.
3. There is a polynomial-time algorithm $\text{SamplePre}(\mathbf{A}, \mathbf{T}_A, \mathbf{u}, \sigma)$ that returns $\mathbf{x} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$ sampled from a distribution statistically close to $D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}), \sigma, \mathbf{c}}$.

Definition 2. For any m -dimensional lattice Λ and positive real $\epsilon > 0$, the smoothing parameter η_ϵ is the smallest real $\sigma > 0$ such that $\rho_{1/\sigma}(\Lambda^* \setminus \{0\}) \leq \epsilon$.

Lemma 3 [27]. Let $\Lambda \subseteq \mathbb{Z}^m$ be a lattice and $\sigma \in \mathbb{R}$. For $i = 1, \dots, k$, $\mathbf{v}_i \in \mathbb{Z}^m$ and let X_i be mutually independent random variables sampled from $D_{\Lambda + \mathbf{v}_i, \sigma}$. Let $\mathbf{c} = (c_1, \dots,$

$c_k) \in \mathbb{Z}^k$, and define $g := \gcd(c_1, \dots, c_k)$, and $\mathbf{v} := \sum_{i=1}^k c_i \mathbf{v}_i$. Suppose that $\sigma > \|\mathbf{c}\| \cdot \eta_\epsilon(\Lambda)$ for some negligible ϵ . Then $Z = \sum_{i=1}^k c_i X_i$ is statistically close to $D_{g\Lambda + \mathbf{v}, \|\mathbf{c}\|\sigma}$.

Definition 3. We say that a matrix A in $\mathbb{Z}^{m \times m}$ is \mathbb{Z}_q -invertible if $A \bmod q$ is invertible as a matrix in $\mathbb{Z}_q^{m \times m}$.

Algorithm 1. [16] $\text{SampleS}(1^m)$

Let $\sigma_s = O(\sqrt{n \log q}) \cdot \omega(\log m) \cdot \sqrt{m}$

1. Let \mathbf{T}_0 be the canonical basis of the lattice \mathbb{Z}^m ;
2. For $i = 1, \dots, m$ do $s_i \leftarrow^R \text{SampleGaussian}(\mathbb{Z}^m, \mathbf{T}_0, \sigma_s, \mathbf{0})$;
3. If \mathbf{S} is \mathbb{Z}_q -invertible, output \mathbf{S} ; otherwise repeat step 2.

2.4 The SIS Problem

In this section, we recall the Small Integer Solution problem, which is essentially the knapsack problem over elements in \mathbb{Z}_q^n . We focus on $l_2 - \text{SIS}_{q,n,m,\beta}$ problem.

Definition 4 ($l_2 - \text{SIS}_{q,n,m,\beta}$ problem). Given an integer q , a random matrix $A \in \mathbb{Z}_q^{n \times m}$ and a real β , find a vector $\mathbf{v} \in \mathbb{Z}^m \setminus \{0\}$ such that $A\mathbf{v} = \mathbf{0} \bmod q$ and $\|\mathbf{v}\| \leq \beta$.

The following lemma shows that $l_2 - \text{SIS}_{q,n,m,\beta}$ problem is as hard as approximating certain worst-case problems on lattice.

Lemma 4 [10]. For any poly-bounded m , $\beta = \text{poly}(n)$ and for any prime $q \geq \beta \cdot \omega(\sqrt{n \log n})$, the average-case problem $l_2 - \text{SIS}_{q,n,m,\beta}$ is as hard as approximating the SIVP problem in the worst-case to within certain $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$.

Lemma 5 [16]. Let $q > 2$, $m > 2n \log q$ and $\sigma > \|\tilde{\mathbf{T}}_A\| \cdot \omega(\sqrt{\log 2m})$. Then there exists a polynomial-time algorithm $\text{SampleBasisLeft}(A, \mathbf{M}, \mathbf{T}_A)$ takes $A, \mathbf{M} \in \mathbb{Z}_q^{n \times m}$ and a basis \mathbf{T}_A of $\Lambda_q^\perp(A)$ as inputs, outputs a basis \mathbf{T}_F of $\Lambda_q^\perp(F)$ with $\|\tilde{\mathbf{T}}_A\| = \|\tilde{\mathbf{T}}_F\|$, where $F = (A|\mathbf{M})$.

3 Proxy Re-Signature: Definition and Security Model

3.1 Definition of Unidirectional Proxy Re-Signature

In this section we recall the definition of the unidirectional proxy re-signature schemes. The unidirectional proxy re-signature scheme for L levels consists of five algorithms (KeyGen, ReKeyGen, Sign, ReSign, Verify)

KeyGen: This algorithm takes as input a security parameter n and returns a user's private/public key pair (sk, pk) .

ReKeyGen: This algorithm takes as input user i 's public key pk_i , user j 's private key sk_j and returns a re-signature key $rk_{i \rightarrow j}$ that allows translating i 's signatures into j 's signatures. The re-signature key $rk_{i \rightarrow j}$ is secret.

Sign: This algorithm takes as input a message μ , a private key sk_i , an integer $l \in [L]$ and returns a signature θ on behalf of user i at level l .

ReSign: This algorithm takes as input public parameters, a level l signature θ for message μ from user i , a re-signature key $rk_{i \rightarrow j}$ and checks that θ is valid. If so, it returns a signature θ' which verifies at level $l+1$ under public key pk_j .

Verify: This algorithm takes as input public parameters, an integer $l \in [L]$, a message μ , a signature θ' , a public key pk_j and returns 0 or 1.

Here, we explain that why the definition contains the level. In a proxy re-signature scheme, if we can distinguish the re-signatures from the original signatures. Without loss of generality, we say that original signatures are the Bob's first-level signatures and the re-signatures are the Bob's second-level signatures. We know that Alice and proxy can produce Bob's re-signatures (second-level signatures). Then it is a secure problem that the first-level signatures are generated by Alice and proxy. If we cannot distinguish the re-signatures from the original signatures, i.e. the first-level signatures and second-level signatures are indistinguishable, the level is not considered.

3.2 Security Model of Unidirectional Proxy Re-Signature

The security model of unidirectional proxy re-signature of [2] considers the following notions termed as external and internal security.

External Security: It is the security against adversaries except the proxy and delegation partners. Formally, for the security parameter n and all probability polynomial time adversaries \mathcal{A} :

$$\begin{aligned} & \Pr[\{(pk_i, sk_i) \leftarrow \text{KeyGen}(1^n)\}_{i \in [1, k]}, \\ & (t, \mu, \theta) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{sign}}(\cdot, \cdot), \mathcal{O}_{\text{resign}}(\cdot, \cdot, \cdot)}(\{pk_i\}_{i \in [1, k]}) : \\ & \text{Verify}(pk_t, \mu, \theta) = 1 \wedge (1 \leq t \leq k) \wedge (t, \mu, \theta) \notin \mathcal{Q}] < 1/\text{poly}(n) \end{aligned} \quad (6)$$

where the oracle $\mathcal{O}_{\text{sign}}$ takes as input an index $i \in [1, k]$ and a message $\mu \in M$ and outputs a signature $\theta \leftarrow \text{Sign}(sk_j, \mu)$. The oracle $\mathcal{O}_{\text{resign}}$ takes as input two distinct indices $1 \leq i, j \leq k$, a message μ and a signature θ and outputs a re-signature $\theta' \leftarrow \text{ReSign}(rk_{i \rightarrow j}, pk_i, \theta, \mu)$. Let \mathcal{Q} denotes the set of tuples (t, μ, θ) where \mathcal{A} obtained a signature θ on μ under public key pk_t by querying $\mathcal{O}_{\text{sign}}$ on (t, μ) or $\mathcal{O}_{\text{resign}}(\cdot, t, \mu, \cdot)$.

Internal Security: This security model can be against the collusion attack (dishonest proxies and colluding delegation partners). The model contains three security guarantees.

1. **Limited Proxy:** This notion protects the honest delegator and delegatee, namely, the proxy can not forge the signatures of the delegatee or delegator unless the message was first signed by one of the latter's delegates. Formally, for the security parameter n and all probability polynomial time adversaries \mathcal{A} :

$$\begin{aligned}
 & \Pr[\{(pk_i, sk_i) \leftarrow \text{KeyGen}(1^n)\}_{i \in [1, k]}, \\
 & \quad (t, \mu, \theta) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{sign}}(\cdot, \cdot), \mathcal{O}_{\text{rekey}}(\cdot, \cdot)}(\{pk_i\}_{i \in [1, k]}): \\
 & \quad \text{Verify}(pk_t, \mu, \theta) = 1 \wedge (1 \leq t \leq k) \wedge (t, \mu) \notin \mathcal{Q}] < 1/\text{poly}(n)
 \end{aligned} \tag{7}$$

where the oracle $\mathcal{O}_{\text{sign}}$ takes as input an index $i \in [1, k]$ and a message $\mu \in M$ and outputs a signature $\theta \leftarrow \text{Sign}(sk_i, \mu)$. The oracle $\mathcal{O}_{\text{rekey}}$ takes as input two distinct indices $1 \leq i, j \leq k$ and outputs the re-signature key $rk_{i \rightarrow j} \leftarrow \text{ReKey}(pk_i, pk_j, sk_j)$. Let \mathcal{Q} denotes the set of tuples (t, μ) where \mathcal{A} obtained a signature on μ under public key pk_t or one of its delegate key's by querying $\mathcal{O}_{\text{sign}}$.

2. **Delegatee Security:** This notion protects the delegatee, i.e., it can be against the collusion attack from delegator and proxy. We associate the index 0 to the delegatee. Formally, for the security parameter n and all probability polynomial time adversaries \mathcal{A} :

$$\begin{aligned}
 & \Pr[\{(pk_i, sk_i) \leftarrow \text{KeyGen}(1^n)\}_{i \in [1, k]}, \\
 & \quad (\mu, \theta) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{sign}}(0, \cdot), \mathcal{O}_{\text{rekey}}(\cdot, \star)}(pk_0, \{pk_i, sk_i\}_{i \in [1, k]}): \\
 & \quad \text{Verify}(pk_0, \mu, \theta) = 1 \wedge (\mu, \theta) \notin \mathcal{Q}] < 1/\text{poly}(n)
 \end{aligned} \tag{8}$$

where $\star \neq 0$ and \mathcal{Q} is the set of pairs (μ, θ) such that \mathcal{A} queried $\mathcal{O}_{\text{sign}}(0, \mu)$ and obtained θ .

3. **Delegator Security:** This notion protects the delegator, i.e., it can be against the collusion attack from delegatee and proxy. That is, there are distinguishable signatures for a user based on whether she used her strong secret key or her weak secret key. The colluding delegatee and proxy cannot produce strong signatures (first-level signature) on her behalf. We associate the index 0 to the delegator. Formally, for the security parameter n and all probability polynomial time adversaries \mathcal{A} :

$$\begin{aligned}
 & \Pr[\{(pk_i, sk_i) \leftarrow \text{KeyGen}(1^n)\}_{i \in [1, k]}, \\
 & \quad (\mu, \theta) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{sign}}(0, \cdot), \mathcal{O}_{\text{rekey}}(\cdot, \cdot)}(pk_0, \{pk_i, sk_i\}_{i \in [1, k]}): \\
 & \quad \text{Verify}(pk_0, \mu, \theta) = 1 \wedge (\mu, \theta) \notin \mathcal{Q}] < 1/\text{poly}(n)
 \end{aligned} \tag{9}$$

where θ is a first-level signature and \mathcal{Q} is the set of pairs (μ, θ) such that \mathcal{A} queried $\mathcal{O}_{\text{sign}}(0, \mu)$ and obtained θ .

4 Multi-use Unidirectional Proxy Re-Signature Scheme from Lattice in the Random Oracle Model

4.1 Our Construction

In this section, we use the Gentry, Peikert, and Vaikuntanathan's signature scheme [10] to construct a multi-use unidirectional proxy re-signature scheme. Let n be a security parameter, and $q \geq \beta \cdot \omega(\log n)$ for $\beta = \text{poly}(n)$. Let $m \geq 2n \log q$ and a Gaussian parameter $\sigma \geq O(\sqrt{n \log q}) \cdot \omega(\sqrt{\log n})$. There is a collision-resistant secure hash function H that maps $\{0, 1\}^*$ to \mathbb{Z}_q^n . Our scheme consists of the following algorithms.

KeyGen: On input the security parameter n , run $\text{TrapGen}(q, n)$ to generate a random rank n matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a trapdoor basis \mathbf{T} of $\Lambda_q^\perp(\mathbf{A})$ such that $\|\tilde{\mathbf{T}}\| \leq O(\sqrt{n \log q})$. Let the trapdoor function $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$. The public key is $pk = \mathbf{A}$, the secret key is $sk = \mathbf{T}$.

Re-Signature Key Generation: On input public keys of user A and B , $pk_A = \mathbf{A}$, $pk_B = \mathbf{B}$ and a secret key $sk_B = \mathbf{T}_B$. Let $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m)^T$, where $\mathbf{a}_i \in \mathbb{Z}_q^n$. For every \mathbf{a}_i , $i = 1, 2, \dots, m$, use preimage sampleable algorithm $\text{SamplePre}(\mathbf{B}, \mathbf{T}_B, \mathbf{a}_i, \sigma)$ which samples a vector s_i such that $\mathbf{B}s_i = \mathbf{a}_i \bmod q$ and $\|s_i\| \leq \sigma\sqrt{m}$. Let $\mathbf{S}_{A \rightarrow B} = (s_1, s_2, \dots, s_m) \in \mathbb{Z}^{m \times m}$, then $\mathbf{B}\mathbf{S}_{A \rightarrow B} = \mathbf{A} \bmod q$ and $\|\mathbf{S}_{A \rightarrow B}\| \leq \sigma\sqrt{m}$. Output the re-signature key $rk_{A \rightarrow B} = \mathbf{S}_{A \rightarrow B}$.

Sign: The first-level signature: on input a secret key $sk = \mathbf{T}$ and a message μ , do:

1. Choose a random vector $r \in \{0, 1\}^*$ and compute $\mathbf{u} = H(\mu \| r) \in \mathbb{Z}_q^n$;
2. Use preimage sampleable algorithm $\text{SamplePre}(\mathbf{A}, \mathbf{T}, \mathbf{u}, \sigma)$ samples a vector \mathbf{e} such that $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$ and $\|\mathbf{e}\| \leq s\sqrt{m}$.
3. Output (\mathbf{e}, r) as the signature for message μ .

The i -level signature: on input a secret key $sk = \mathbf{T}$ and a message μ , do:

4. Choose a random vector $r \in \{0, 1\}^*$ and compute $\mathbf{u} = H(\mu \| r) \in \mathbb{Z}_q^n$;
5. Use preimage sampleable algorithm $\text{SamplePre}(\mathbf{A}, \mathbf{T}, \mathbf{u}, \sigma^i m^{(i-1)/2})$ to sample a vector \mathbf{e} such that $\mathbf{A}\mathbf{e} = \mathbf{u} \bmod q$ and $\|\mathbf{e}\| \leq \sigma^i m^{i/2}$.
6. Output (\mathbf{e}, r) as the signature for message μ .

Re-Signature: On input re-signature key $rk_{A \rightarrow B} = \mathbf{S}_{A \rightarrow B}$, a public key $pk_A = \mathbf{A}$, a message μ and a first-level signature (\mathbf{e}_A, r) , check that $\mathbf{A}\mathbf{e}_A = \mathbf{u} \bmod q$ and $\|\mathbf{e}_A\| \leq \sigma\sqrt{m}$. If \mathbf{e}_A is not a signature for μ , output \perp ; otherwise compute re-signature $\mathbf{e}_B = \mathbf{S}_{A \rightarrow B}\mathbf{e}_A$. (\mathbf{e}_B, r) is the re-signature for $A \rightarrow B$.

The algorithm ReSign can transform an l -level signature into $(l + 1)$ -level signature as first-level re-signature.

Verify: On input a public key $pk_B = \mathbf{B}$, a message μ and a re-signature (\mathbf{e}_B, r) for $A \rightarrow B$. If $\mathbf{B}\mathbf{e}_B = \mathbf{u} \bmod q$ and $\|\mathbf{e}_B\| \leq \sigma^2 m$, output 1; otherwise output 0.

4.2 Security and Good Properties

Theorem 1 (Multi-use). The scheme is multi-use correct.

Proof: Consider the users $1, \dots, k$. Suppose (e_1, r) is a valid signature of user 1, i.e., $A_1 e_1 = H(\mu || r) \bmod q$ and $\|e_1\| \leq \sigma\sqrt{m}$. Re-signature procedure is performed from 1 to k through 2 to $k-1$. The re-signature procedure is as follows:

$$\begin{aligned} e_k &= S_{k-1 \rightarrow k} e_{k-1} = S_{k-1 \rightarrow k} S_{k-2 \rightarrow k-1} e_{k-2} \\ &= \dots = S_{k-1 \rightarrow k} S_{k-2 \rightarrow k-1} \dots S_{2 \rightarrow 1} e_1 \end{aligned} \quad (10)$$

The verification procedure by the public key A_k of user k is as follows:

$$\begin{aligned} A_k e_k &= A_k S_{k-1 \rightarrow k} S_{k-2 \rightarrow k-1} \dots S_{2 \rightarrow 1} e_1 \\ &= A_{k-1} S_{k-2 \rightarrow k-1} \dots S_{2 \rightarrow 1} e_1 \\ &= A_1 e_1 \\ &= u \bmod q \end{aligned} \quad (11)$$

and

$$\begin{aligned} \|e_k\| &= \|S_{k-1 \rightarrow k} S_{k-2 \rightarrow k-1} \dots S_{2 \rightarrow 1} e_1\| \\ &\leq \|S_{k-1 \rightarrow k}\| \dots \|S_{2 \rightarrow 1}\| \|e_1\| \\ &\leq \sigma^k m^{k/2} \end{aligned} \quad (12)$$

Therefore, the scheme is multi-use correct.

In the following, we analyze the other properties.

Theorem 2. In a random oracle model, the scheme is secure under the $SIS_{q,n,m,\beta}$ problem, more precisely, given a random rank n matrix $A \in \mathbb{Z}_q^{n \times m}$, if finding a non-zero vector v such that $Av = \mathbf{0} \bmod q$ and $\|v\| \leq \beta$ is hard, then the scheme is secure.

Proof: We argue security in two parts, i.e., the external security and the internal security.

External Security: For security, we assume there is a probability poly-time adversary \mathcal{A} which breaks this guarantee with non-negligible probability ε after making at most q_H hash queries, q_s signature queries and q_{rs} re-signature queries. We use \mathcal{A} to construct a poly-time simulator \mathcal{B} that solves the $SIS_{q,n,m,\beta}$ problem.

System Parameters: On input a random matrix $A \in \mathbb{Z}_q^{n \times m}$, the simulator \mathcal{B} outputs a non-zero vector v such that $Av = \mathbf{0} \bmod q$ and $\|v\| \leq \beta$.

Public keys: When \mathcal{A} asks for the creation of user $i \in \{1, \dots, \kappa\}$, \mathcal{B} needs to prepare κ public keys $\mathbf{A}_1, \dots, \mathbf{A}_\kappa$. The procedure is as follows:

- (i) Let $\mathbf{A} = \mathbf{A}_t$. \mathcal{B} uses the algorithm $\text{SampleS}(1^m)$ to sample $t-1$ matrices $\mathbf{S}_{t-1 \rightarrow t}, \dots, \mathbf{S}_{1 \rightarrow 2}$ and computes $\mathbf{A}_{t-1} = \mathbf{A}_t \mathbf{S}_{t-1 \rightarrow t} \bmod q, \dots, \mathbf{A}_1 = \mathbf{A}_2 \mathbf{S}_{1 \rightarrow 2} \bmod q$.
- (ii) \mathcal{B} uses $\text{TrapGen}(1^n)$ to generate $\kappa-t$ public/secret key pairs $(\mathbf{A}_i, \mathbf{T}_i)$, $i = t+1, \dots, \kappa$.

In the following, \mathcal{B} must answer the random oracle H , the signature oracle $\mathcal{O}_{\text{sign}}$ and the re-signature oracle $\mathcal{O}_{\text{resign}}$. \mathcal{B} simulates these oracles as follows:

Hash queries: \mathcal{B} maintains a list of tuples $(i, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ which is called the H list. For each query to H , if (μ_k, r_k) is in the H list, then \mathcal{B} returns \mathbf{u}_k to \mathcal{A} . Otherwise, if $i > t$, compute $\mathbf{u}_k = H(\mu_k || r_k)$ and use the secret key \mathbf{T}_i to sample a vector $\mathbf{e}_k \leftarrow \text{SamplePre}(\mathbf{A}_i, \mathbf{T}_i, \mathbf{u}_k, \sigma_i)$, store $(i, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ and return \mathbf{u}_k to \mathcal{A} . If $i \leq t$, sample $\mathbf{e}_k \leftarrow D_{\mathbb{Z}^m, s_i}$ and compute $\mathbf{u}_k = \mathbf{A}_i \mathbf{e}_k \bmod q$, store $(i, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ and return \mathbf{u}_k to \mathcal{A} .

Signature queries: For each query to $\mathcal{O}_{\text{sign}}$ on input $(i, (\mu_k, r_k))$. We assume that μ_k has already been queried on the random oracle H . \mathcal{B} looks up $(i, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ in the H list and returns \mathbf{e}_k to \mathcal{A} .

Re-Signature queries: For each query to $\mathcal{O}_{\text{resign}}$ on input $(i, j, (\mu_k, r_k), \mathbf{e}_k)$, if $j > t$, compute re-signature key $rk_{i \rightarrow j} = \mathbf{S}_{i \rightarrow j}$ by the *Re-Signature key generation algorithm* and compute $\mathbf{e}'_k = \mathbf{S}_{i \rightarrow j} \mathbf{e}_k$, and then return \mathbf{e}'_k to \mathcal{A} . Otherwise, if $j \leq t$, compute $rk_{i \rightarrow j} = \mathbf{S}_{i \rightarrow j} = \mathbf{S}_{j-1 \rightarrow j} \cdots \mathbf{S}_{i \rightarrow i+1}$ and $\mathbf{e}'_k = \mathbf{S}_{i \rightarrow j} \mathbf{e}_k$, and then return \mathbf{e}'_k to \mathcal{A} .

Forgery: Without loss of generality, we assume that \mathcal{A} selects \mathbf{A}_t as the challenge public key (the probability is $1/\kappa$) before outputting its forgery $((\mu^*, r^*), \mathbf{e}^*)$ and querying H on μ^* . Finally, \mathcal{A} outputs forgery $((\mu^*, r^*), \mathbf{e}^*)$.

We now analyze the simulation. First, for each distinct query (μ, r) to H , the value \mathbf{u} returned by \mathcal{B} is $\mathbf{u} = f_{\mathbf{A}}(\mathbf{e}) = \mathbf{A} \mathbf{e} \bmod q$, where $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, s}$. Because the distribution of \mathbf{u} is uniform, it is identical to the uniformly random value of $H(\mu || r)$ in the real system. Second, for each query (μ, r) to $\mathcal{O}_{\text{sign}}$, \mathcal{B} returns a single value $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, s}$ such that $f_{\mathbf{A}}(\mathbf{e}) = H(\mu || r)$. In the real system, signature queries on μ are answered by a single value with the same distribution by the algorithm SamplePre . Third, for each query to $\mathcal{O}_{\text{resign}}$, we know that the re-signature key in $\mathcal{O}_{\text{resign}}$ queries is indistinguishable from that in the real system, so the $\mathcal{O}_{\text{resign}}$ queries is statistically close to the view of the real system. Thus we claim that the simulation of \mathcal{B} is identical to the real system.

When \mathcal{A} outputs forgery $((\mu^*, r^*), \mathbf{e}^*)$, \mathcal{B} looks up $((\mu^*, r^*), \mathbf{e}_{\mu^*})$ in the H list and outputs $\mathbf{v} = \mathbf{e}_{\mu^*} - \mathbf{e}^*$ as the solution of the $\text{SIS}_{q,n,m,\beta}$ problem $\mathbf{A} \mathbf{v} = \mathbf{0} \bmod q$. Because $((\mu^*, r^*), \mathbf{e}^*)$ and $((\mu^*, r^*), \mathbf{e}_{\mu^*})$ are both the signatures of μ^* , then

$$\mathbf{A}_t \mathbf{e}^* \bmod q = H(\mu^* || r^*) \bmod q = \mathbf{A}_t \mathbf{e}_{\mu^*} \bmod q \quad (13)$$

Therefore, we obtain $\mathbf{A}_t(\mathbf{e}^* - \mathbf{e}_{\mu^*}) = \mathbf{0} \bmod q$. Since $\|\mathbf{e}^*\|, \|\mathbf{e}_{\mu^*}\| \leq \sigma\sqrt{m}$ and $\mathbf{e}^* \neq \mathbf{e}_{\mu^*}$, we have $\|\mathbf{e}^* - \mathbf{e}_{\mu^*}\| \leq 2\sigma\sqrt{m}$ and $\mathbf{e}^* - \mathbf{e}_{\mu^*} \neq \mathbf{0}$.

Internal Security: In this scheme, since the first-level signatures belong to the second-level signatures, the colluding delegatee and proxy can produce a first-level signature on delegator's behalf. Thus, the delegator security in our scheme is not satisfied. Internal security refers only to the limited proxy security and delegatee security.

Limited Proxy Security: For security, we assume there is a probability poly-time adversary (proxy) \mathcal{A} which breaks this guarantee with non-negligible probability. We use \mathcal{A} to construct a poly-time simulator \mathcal{B} that solves the $\text{SIS}_{q,n,m,\beta}$ problem.

System Parameters: On input a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the simulator \mathcal{B} outputs a non-zero vector \mathbf{v} such that $\mathbf{A}\mathbf{v} = \mathbf{0} \bmod q$ and $\|\mathbf{v}\| \leq \beta$.

Public keys: When \mathcal{A} asks for the creation of user $i \in \{1, \dots, \kappa\}$, \mathcal{B} needs to prepare κ public keys $\mathbf{A}_1, \dots, \mathbf{A}_\kappa$. The procedure is as follows:

- (i) \mathcal{B} sets $\mathbf{A} = \mathbf{A}_t$.
- (ii) \mathcal{B} uses $\text{TrapGen}(1^n)$ to generate $\kappa - 1$ pairs of public/secret keys $(\mathbf{A}_i, \mathbf{T}_i)$, $i = 1, \dots, t - 1, \dots, t + 1, \dots, \kappa$.

In the following, \mathcal{B} must answer the random oracle H , the signature oracle $\mathcal{O}_{\text{sign}}$ and the re-signature key oracle \mathcal{O}_{rk} . \mathcal{B} simulates these oracles as follows:

Hash queries: \mathcal{B} maintains a list of tuples $(i, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ which is called the H list. for each query to H , if (μ_k, r_k) is in the H list, then \mathcal{B} returns \mathbf{u}_k to \mathcal{A} . Otherwise, if $i \neq t$, choose a random vector $r_k \in \{0, 1\}^*$, compute $\mathbf{u}_k = H(\mu_k \| r_k)$ and use the secret key \mathbf{T}_i to sample a vector $\mathbf{e}_k \leftarrow \text{SamplePre}(\mathbf{A}_i, \mathbf{T}_i, \mathbf{u}_k, \sigma_i)$, store $(i, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ and return \mathbf{u}_k to \mathcal{A} . If $i = t$, sample $\mathbf{e}_k \leftarrow D_{\mathbb{Z}^m, s}$ and compute $\mathbf{u}_k = \mathbf{A}_i \mathbf{e}_k \bmod q$, store $(i, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ and return \mathbf{u}_k to \mathcal{A} .

Signature queries: For each query to $\mathcal{O}_{\text{sign}}$ on input $(i, (\mu_k, r_k))$. We assume that μ_k has already been queried on the random oracle H . \mathcal{B} looks up $(i, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ in the H list and returns \mathbf{e}_k to \mathcal{A} .

Re-Signature key queries: For each query to \mathcal{O}_{rk} on input (i, j) , if $i = t$ or $j = t$, abort; otherwise, compute re-signature key $rk_{i \rightarrow j} = \mathbf{S}_{i \rightarrow j}$ by the *Re-Signature key generation algorithm* and return $rk_{i \rightarrow j} = \mathbf{S}_{i \rightarrow j}$ to \mathcal{A} .

Forgery: Without loss of generality, we assume that \mathcal{A} selects \mathbf{A}_t as the challenge public key (the probability is $1/\kappa$) before outputting its forgery $((\mu^*, r^*), \mathbf{e}^*)$ and querying H on μ^* . Finally, \mathcal{A} outputs forgery $((\mu^*, r^*), \mathbf{e}^*)$.

Simulator \mathcal{B} 's simulation of the world for \mathcal{A} is the same as the external security except that the Re-Signature queries is replaced by the Re-Signature key queries.

Delegatee security: For security, we assume there is a probability poly-time adversary (proxy) \mathcal{A} which breaks this guarantee with non-negligible probability. We use \mathcal{A} to construct a poly-time simulator \mathcal{B} that solves the $\text{SIS}_{q,n,m,\beta}$ problem.

System Parameters: On input a random matrix $A \in \mathbb{Z}_q^{n \times m}$, the simulator \mathcal{B} outputs a non-zero vector \mathbf{v} such that $A\mathbf{v} = \mathbf{0} \bmod q$ and $\|\mathbf{v}\| \leq \beta$.

Public keys: When \mathcal{A} asks for the creation of user $i \in \{1, \dots, \kappa\}$, \mathcal{B} needs to prepare κ public keys A_1, \dots, A_κ . The procedure is as follows:

- (i) \mathcal{B} sets $A = A_1$.
- (ii) \mathcal{B} uses $TrapGen(1^n)$ to generate $k - 1$ pairs of public/secret keys (A_i, T_i) , $i = 2, \dots, \kappa$.

In the following, \mathcal{B} must answer the random oracle H , the signature oracle \mathcal{O}_{sign} and the re-signature key oracle \mathcal{O}_{rk} . \mathcal{B} simulates these oracles as follows:

Hash queries: \mathcal{B} maintains a list of tuples $(i, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ which is called the H list. for each query to H , if μ_k is in the H list, \mathcal{B} returns \mathbf{u}_k to \mathcal{A} . Otherwise, if $i \neq 1$, choose a random vector $r_k \in \{0, 1\}^*$, compute $\mathbf{u}_k = H(\mu_k \| r_k)$ and use the secret key T_i to sample a vector $\mathbf{e}_k \leftarrow SamplePre(A_i, T_i, \mathbf{u}_k, \sigma_i)$, store $(i, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ and return \mathbf{u}_k to \mathcal{A} . If $i = 1$, sample $\mathbf{e}_k \leftarrow D_{\mathbb{Z}_q^m, s}$ and compute $\mathbf{u}_k = A_1 \mathbf{e}_k \bmod q$, store $(1, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ and return \mathbf{u}_k to \mathcal{A} .

Signature queries: For each query to \mathcal{O}_{sign} on input $(i, (\mu_k, r_k))$. We assume that μ_k has already been queried on the random oracle H . \mathcal{B} looks up $(i, \mathbf{u}_k, \mathbf{e}_k, (\mu_k, r_k))$ in the H list and returns \mathbf{e}_k to \mathcal{A} .

Re-Signature key queries: For each query to \mathcal{O}_{rk} on input (i, j) , if $i = 1$, abort; otherwise, compute re-signature key $rk_{i \rightarrow j} = S_{i \rightarrow j}$ by the *Re-Signature key generation algorithm* and return $rk_{i \rightarrow j} = S_{i \rightarrow j}$ to \mathcal{A} .

Forgery: Without loss of generality, we assume that \mathcal{A} selects A_i as the challenge public key (the probability is $1/\kappa$) before outputting its forgery $((\mu^*, r^*), \mathbf{e}^*)$ and querying H on μ^* . Finally, \mathcal{A} outputs forgery $((\mu^*, r^*), \mathbf{e}^*)$.

We know that the simulation is perfect. When \mathcal{A} outputs forgery $((\mu^*, r^*), \mathbf{e}^*)$, \mathcal{B} looks up $((\mu^*, r^*), \mathbf{e}_{\mu^*})$ in the H list and outputs $\mathbf{v} = \mathbf{e}_{\mu^*} - \mathbf{e}^*$ as the solution of the $SIS_{q,n,m,\beta}$ problem $A\mathbf{v} = \mathbf{0} \bmod q$. Because $((\mu^*, r^*), \mathbf{e}^*)$ and $((\mu^*, r^*), \mathbf{e}_{\mu^*})$ are both the signatures of μ^* , then

$$A_1 \mathbf{e}^* \bmod q = H(\mu^* \| r^*) \bmod q = A_1 \mathbf{e}_{\mu^*} \bmod q \quad (14)$$

Therefore, we obtain $A_1(\mathbf{e}^* - \mathbf{e}_{\mu^*}) = \mathbf{0} \bmod q$. Since $\|\mathbf{e}^*\|, \|\mathbf{e}_{\mu^*}\| \leq \sigma\sqrt{m}$ and $\mathbf{e}^* \neq \mathbf{e}_{\mu^*}$, we have $\|\mathbf{e}^* - \mathbf{e}_{\mu^*}\| \leq 2\sigma\sqrt{m}$ and $\mathbf{e}^* - \mathbf{e}_{\mu^*} \neq \mathbf{0}$.

4.3 Security and Efficiency Comparison

In this section, we compare the security and efficiency of the proposed scheme with that of the scheme of [3] which is the first multi-use unidirectional proxy re-signature scheme. The scheme needs 6 pair operations in the verification of 1-level signature, and $4L + 2$ pair operations in the verification of L -level signature. The proposed

construction is based on the Small Integer Solution problem. The verification cost does not grow with the number of translations (only one matrix-vector product operation in any level signature) and the size of signatures also grows linearly with the number of translations. The comparison results are summarized in Table 1.

Table 1. Security and efficiency comparison

Cryptosystem	Underlying problem	The size of signature	Verification cost
The scheme of [3]	l -FlexDH assumption	Grows linearly with the number of translations	Grows linearly with the number of translations
The proposed scheme	SIS problem	Grows linearly with the number of translations	Not change with the number of translations

5 Multi-use Unidirectional Proxy Re-Signature Scheme from Lattice in the Standard Model

In this section, we use the signature scheme of [15] to construct a multi-use unidirectional proxy re-signature scheme in the standard model.

KeyGen: On input the security parameter n , run $TrapGen(q, n)$ to generate a random rank n matrix $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ and a trapdoor basis \mathbf{T}_0 of $\Lambda_q^\perp(\mathbf{A}_0)$ such that $\|\tilde{\mathbf{T}}_0\| \leq O(\sqrt{n \log q})$.

For each $(b, j) \in \{0, 1\} \times [k]$, choose uniformly random and independent $\mathbf{A}_j^{(b)} \in \mathbb{Z}_q^{n \times m}$. Output public key $pk = (\mathbf{A}_0, \mathbf{A}_j^{(b)})$ and secret key $sk = \mathbf{T}_0$.

Re-Signature Key Generation: On input public keys of user 1 and 2, $pk_1 = (\mathbf{A}_{10}, \mathbf{A}_j^{(b)})$, $pk_2 = (\mathbf{A}_{20}, \mathbf{A}_j^{(b)})$ and a secret key $sk_2 = \mathbf{T}_2$. Let $\mathbf{A}_{10} = (\mathbf{a}_{11}, \mathbf{a}_{12}, \dots, \mathbf{a}_{1m})^T$, where $\mathbf{a}_{1i} \in \mathbb{Z}_q^n$. For every \mathbf{a}_{1i} , $i = 1, 2, \dots, m$, use preimage sampleable algorithm $SamplePre(\mathbf{A}_{20}, \mathbf{T}_2, \mathbf{a}_{1i}, \sigma)$ which samples a vector \mathbf{s}_i such that $\mathbf{A}_{20}\mathbf{s}_i = \mathbf{a}_{1i} \bmod q$ and $\|\mathbf{s}_i\| \leq \sigma\sqrt{m}$. Let $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m) \in \mathbb{Z}^{m \times m}$, then $\mathbf{A}_{20}\mathbf{S} = \mathbf{A}_{10} \bmod q$ and $\|\mathbf{S}\| \leq s\sqrt{m}$. Let $\mathbf{S}_{1 \rightarrow 2} = \begin{pmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$ and output the re-signature key $rk_{1 \rightarrow 2} = \mathbf{S}_{1 \rightarrow 2}$.

Sign: The first-level signature: on input a secret key $sk = \mathbf{T}_0$ and a message $\mu \in \{0, 1\}^k$, do:

1. Let $\mathbf{A}_\mu = \mathbf{A}_0 \|\mathbf{A}_1^{(\mu_1)}\| \dots \|\mathbf{A}_k^{(\mu_k)}\| \in \mathbb{Z}_q^{n \times (k+1)m}$. Use $SampleBasisLeft(\mathbf{A}_0, \mathbf{A}_i^{(\mu_i)}, \mathbf{T}_0)$ to generate the basis \mathbf{T}_μ of $\Lambda^\perp(\mathbf{A}_\mu)$;
2. Use preimage sampleable algorithm $SamplePre(\mathbf{A}_\mu, \mathbf{T}_\mu, \mathbf{0}, \sigma)$ to sample a vector \mathbf{e} such that $\mathbf{A}_\mu \mathbf{e} = \mathbf{0} \bmod q$ and $\|\mathbf{e}\| \leq \sigma\sqrt{(k+1)m}$.
3. Output \mathbf{e} as the signature for message μ .

The i -level signature: on input a secret key $sk = T_0$ and a message μ , do:

1. Let $A_\mu = A_0 \| A_1^{(\mu_1)} \| \dots \| A_k^{(\mu_k)} \in \mathbb{Z}_q^{n \times (k+1)m}$. Use *SampleBasisLeft*($A_0, A_i^{(\mu_i)}, T_0$) to generate the basis T_μ of $\Lambda^\perp(A_\mu)$;
2. Use preimage sampleable algorithm *SamplePre*($A_\mu, T_\mu, \mathbf{0}, \sigma^i[(k+1)m]^{(i-1)/2}$) to sample a vector e such that $A_\mu e = \mathbf{0} \pmod q$ and $\|e\| \leq \sigma^i[(k+1)m]^{i/2}$.
3. Output e as the i -level signature for message μ .

Re-Signature: On input re-signature key $rk_{1 \rightarrow 2} = S_{1 \rightarrow 2}$, a public key $pk_1 = (A_{10}, A_j^{(b)})$, a message μ and its signature e_1 , check that $A_{1\mu} e_1 = \mathbf{0} \pmod q$ and $\|e_1\| \leq s\sqrt{(k+1)m}$, where $A_{1\mu} = A_{10} \| A_1^{(\mu_1)} \| \dots \| A_k^{(\mu_k)} \in \mathbb{Z}_q^{n \times (k+1)m}$. If e_1 is not a signature for μ , output \perp ; otherwise compute re-signature $e_2 = S_{1 \rightarrow 2} e_1$. e_2 is the re-signature for $1 \rightarrow 2$.

Verify: On input a public key $pk_2 = (A_{20}, A_j^{(b)})$, a message μ and a re-signature e_2 for $1 \rightarrow 2$. If $A_{2\mu} e_2 = \mathbf{0} \pmod q$ and $\|e_2\| \leq \sigma^2(k+1)m$, where $A_{2\mu} = A_{20} \| A_1^{(\mu_1)} \| \dots \| A_k^{(\mu_k)} \in \mathbb{Z}_q^{n \times (k+1)m}$, output 1; otherwise output 0.

6 Conclusion

In this paper, we construct the first multi-use unidirectional proxy re-signature scheme based on the hardness of the Small Integer Solution (SIS) problem. In our scheme, the verification cost does not grow with the number of translations which only needs a matrix-vector multiplication. The size of signatures grows linearly with the number of the translations in this scheme. Our scheme only uses one signature algorithm such that the user's i -level signatures contain $(i - 1)$ -level signatures, however it does not resist the collusion attack of delegator security.

Acknowledgments. We are thankful to anonymous referees for their helpful comments. This paper is supported by the National Natural Science Foundation of China under Grant No. 61902140, No. 60573026, the Anhui Provincial Natural Science Foundation under Grant No. 1708085QF154, No. 1908085QF288, NO. 1808085QF181, the Nature Science Foundation of Anhui Higher Education Institutions under Grant No. KJ2019A0605, No. KJ2018A0398, No. KJ2019B018.

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. Ateniese, G., Hohenberger, S.: Proxy re-signatures: new definitions, algorithms, and applications. In: CCS Proceedings of the 15th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, March 2005, pp. 310–319 (2005). <https://doi.org/10.1145/1102120.1102161>
3. Libert, B., Vergnaud, D.: Multi-use unidirectional proxy re-signatures. In: CCS Proceedings of the 15th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, October 2008, pp. 511–520 (2008)
4. Sunitha, N.R., Amberker, B.B.: Multi-use unidirectional forward-secure proxy re-signature scheme. In: Proceedings of the 3rd IEEE International Conference on Internet Multimedia Services Architecture and Applications, Bangalore, India, December 2009, pp. 223–228 (2009)
5. Sunitha, N.R.: Proxy re-signature schemes: multi-use, unidirectional and translations. *J. Adv. Inf. Technol.* **2**(3), 165–176 (2011)
6. Shao, J., Cao, Z., Wang, L., Liang, X.: Proxy re-signature schemes without random oracles. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, pp. 197–209. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77026-8_15
7. Shao, J., Wei, G.Y., Ling, Y., Xie, M.D.: Unidirectional identity-based proxy re-signature. In: Proceedings of the IEEE Communications Society, Hangzhou, China, June 2011, pp. 1–5 (2011)
8. Shao, J., Feng, M., Zhu, B., Cao, Z., Liu, P.: The security model of unidirectional proxy re-signature with private re-signature key. In: Steinfeld, R., Hawkes, P. (eds.) ACISP 2010. LNCS, vol. 6168, pp. 216–232. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14081-5_14
9. Yang, P.Y., Cao, Z.F., Dong, X.L.: Threshold proxy re-signature. *J. Syst. Sci. Complex* **2011**(24), 816–824 (2011)
10. Gentry, C., Peikert, C., Vaikuntanathan, V.: How to use a short basis: trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the STOC 2008, Victoria, Canada, May 2008, pp. 197–206 (2008)
11. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. *J. ACM* **56**(6), Article 34 (2009)
12. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_1
13. Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-based encryption. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 319–339. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19074-2_21
14. Stehlé, D., Steinfeld, R.: Making NTRU as secure as worst-case problems over ideal lattices. In: Paterson, Kenneth G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_4
15. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_27
16. 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
17. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_6
18. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the STOC 2009, Bethesda, Maryland, USA, May 2009, pp. 169–178 (2009)

19. Gentry, C.: Toward basing fully homomorphic encryption on worst-case hardness. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 116–137. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_7
20. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_29
21. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Proceedings of the FOCS 2011, Palm Springs, CA, USA, October 2011, pp. 97–106 (2011)
22. Lyubashevsky, V.: lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_43
23. Gordon, S.D., Katz, J., Vaikuntanathan, V.: A group signature scheme from lattice assumptions. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 395–412. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_23
24. Rückert, M.: Lattice-based blind signatures. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 413–430. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_24
25. Rückert, M.: Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles. In: Sendrier, N. (ed.) PQCrypto 2010. LNCS, vol. 6061, pp. 182–200. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12929-2_14
26. Alwen, J., Peiker, C.: Generating shorter bases for hard random lattices. In: Proceedings of the STACS 2009, Freiburg, Germany, February 2009, pp. 75–86 (2009)
27. Boneh, D., Freeman, D.M.: Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 1–16. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_1



Batch Verification of Linkable Ring Signature in Smart Grid

Qiyu Wang¹, Jie Chen^{1,2}(✉), and Lishuang Zhuang¹

¹ State Key Laboratory of Integrated Service Networks, Xidian University,
Xi'an, Shaanxi 710071, China

jchen@mail.xidian.edu.cn

² Cryptographic Research Center, Xidian University, Xi'an 710071, P. R. China

Abstract. In order to realize the secure and efficient transmission of user electricity information in the smart grid, this paper proposes a batch verification of linkable ring signature scheme, which implements batch verification when the selected ring members are the same. The BVLRS scheme is based on batch verification and linkable message tagging and ring signature techniques. Through security analysis, we prove that the scheme is anonymous, unforgeable and linkable under the standard model. The linkable feature guarantees that the district gateway (DGW) calculates the total power consumption of each user without knowing the specific identity of the user, and can also determine the malicious user based on this feature. Through performance analysis, we know that the computational complexity of batch verification of this scheme is: $[4n\eta + 4n + 4\eta + 6]E + [4n + \eta + 4]P$, where E is an exponentiation computation, P is a bilinear pairing computation, n is the number of electricity user, and η is the number of signatures. BVLRS scheme significantly reduces the computational cost of ring signature verification.

Keywords: Smart grid · Ring signature · Batch verification

1 Introduction

With the increasing demand for power resources, traditional power grids have shown many problems in terms of energy efficiency, environmental protection, and security. Due to the existence of these problems, the concept of smart grid is born. Smart grid is a new modernization system with high informationization, automation and interactive features based on traditional power system, through the integration of new energy, new materials, new equipment and advanced sensing technology, information communication technology and automatic control technology. This can better achieve safe, reliable, economical and efficient operation of the grid. According to the smart grid conceptual model [1] proposed by the National Institute of Standards and Technology, the smart grid consists of seven entities: power station, transmission network, distribution network, customer, market, service provider and operation center, in which the first four entities have the characteristics of two-way flow of power and information, and the latter three entities are mainly responsible for information collection and power management of the smart grid. The smart grid utilizes advanced information and

communication technologies to facilitate the intelligent control and economic management of users' electricity, but there is also the problem of user privacy data leakage.

This paper mainly focuses on the privacy protection of information transmitted between electricity users and operation centers, that is, the security requirement of smart grid [2]. At the same time, the smart grid needs to meet the practical requirement, that is, the operation center can calculate the total electricity consumption information of all users. At present, many scholars use data aggregation or anonymity technology to achieve the above two requirements. The proposed scheme based on data aggregation can process a large amount of information without losing basic information, reducing data volume, reducing network redundancy and improving network performance [3–10]. In 2009, Roberto et al. proposed a wireless sensor network data aggregation scheme based on homomorphic encryption [3], which uses symmetric key homomorphic encryption to protect information privacy and completeness. In 2012, Lu et al. proposed a privacy protection aggregation scheme [4], which uses superincreasing sequences to construct multidimensional data and encrypt the structured data by the Paillier cryptosystem technique. In 2013, Sushmita et al. proposed a security strategy [5] combining data combination and access control by using homomorphic encryption technology and attribute-based encryption technology in the smart grid system. In 2015, Erkin proposed a privacy protection data aggregation scheme [6] for smart grids by using the Chinese remainder theorem and homomorphic encryption. In 2017, Shen et al. proposed an efficient cube data aggregation scheme [7] with privacy protection through Horner rule and Paillier homomorphic encryption system. The scheme can distinguish and aggregate different types of data, but can only aggregate electricity user data at the same time point, and cannot aggregate the sum of power consumption data of a single electricity user over a period of time. In 2018, Asmaa et al. proposed a lightweight lattice-based homomorphic privacy-preserving data aggregation scheme [8] for the users in smart grid. In 2018, Lang et al. proposed a Multidimensional-data Tight Aggregation scheme [9] which supports privacy preserving and fine-grained access control. In 2019, Prosanta et al. proposed a lightweight and privacy-friendly masking-based spatial data aggregation scheme [10] for secure forecasting of power demand in smart grids. Next, we analyze the scheme based on anonymity technology [11–17]. In 2011, Cheung et al. adopted the technology of blind signature and anonymous credentials to solve the problem of privacy protection and authentication in power grid [11]. In 2014, Yu et al. adopted the ring signature method to protect the privacy of electricity users [12]. The introduction of ring signature can reduce the system requirements and avoid generating a large number of anonymous credentials. In 2014, Badra et al. proposed a virtual ring structure [13] when protecting the privacy of electricity users. However, due to the characteristics of virtual rings, it is difficult to find malicious users who post fake messages. In 2016, Tan et al. proposed a privacy data collection scheme [14] using pseudonyms in smart grid. The scheme uses ring signature and zero knowledge proof in the process of pseudonym registration. In 2016, Gong et al. proposed a privacy-preserving scheme [15] for incentive-based demand response in the smart grid, which uses the discrete logarithm to create pseudonyms and uses ring signatures to hide user identity during the pseudonym registration process. In 2018, Guan et al. proposed an efficient privacy protection aggregation scheme [16] in the smart grid. The scheme uses the Bloom filter to improve the verification speed of pseudonym verification.

Since the security of pseudonyms is based on the number of pseudonym certificates. Allocating a large number of pseudonyms will lead to large storage cost and waste of pseudonyms. The main cost of ring signature is the computation of signature verification, so we propose the concept of same ring signature with batch verification (i.e. Ring signatures of multiple users can be batch verified if the selection ring remains unchanged). This paper realizes the function of batch verification in ring signature [18] and leads into linkable message tagging [19] technology. Finally, we propose batch verification of linkable ring signature in Smart Grid, and give detailed security proof and performance analysis.

The structure of this paper is as follows. In Sect. 2, we introduce relevant preliminaries and models. Then we propose a BVLRS scheme in Sect. 3 and give the proof of the security of the BVLRS scheme in Sect. 4, followed by the performance analysis of the BVLRS scheme in Sect. 5. Finally, we draw our conclusions in Sect. 6.

2 Preliminaries and Models

2.1 Linkable Message Tagging (LMT) Scheme

A linkable message tagging (LMT) scheme [19] $L = (KGen, Tag, Link)$ and the following efficient algorithms:

$KGen(1^\lambda)$: On input the security parameter λ , this probabilistic algorithm outputs a tagging key tk .

$Tag(tk, m)$: On input a tagging key tk and message m , this algorithm outputs a tag τ .

$Link(m_1, \tau_1, m_2, \tau_2)$: On input message-tag pairs $(m_1, \tau_1), (m_2, \tau_2)$, this deterministic algorithm outputs either 0 or 1.

2.2 Composite Order Bilinear Pairing

Composite order bilinear groups were introduced in [20]. We define them by using a group generator (\mathcal{G}), an algorithm which takes a security parameter (λ) as input and outputs a description of the bilinear group (G). In our case, \mathcal{G} outputs $(N = p_1 p_2 p_3, G, G_T, e)$ where p_1, p_2, p_3 are distinct primes, G and G_T are cyclic groups of order $N = p_1 p_2 p_3$ and $e : G \times G \rightarrow G_T$ is a map such that:

- (1) Bilinearity: For all $g, h \in G$, and $a, b \in \mathbb{Z}_N$, $e(g^a, h^b) = e(g, h)^{ab}$.
- (2) Non-degeneracy: There exists $g, h \in G$ such that $e(g, g)$ has order N in G_T .
- (3) Computability: It is efficient to compute $e(g, h)$ for all $g, h \in G$.

2.3 Complexity Assumptions

We review some complexity assumptions in bilinear groups, which have been defined in [21].

Assumption 1. Give a group generator (\mathcal{G}), we define the following distribution: $G = (N = p_1 p_2 p_3, G, G_T, e) \leftarrow_R \mathcal{G}$, $g \leftarrow_R G_{p_1}$, $X_3 \leftarrow_R G_{p_3}$, $D = (G, g, X_3)$, $T_1 \leftarrow_R G_{p_1 p_2}$, $T_2 \leftarrow G_{p_1}$. We define the advantage of an algorithm (\mathcal{A}) in breaking Assumption 1 to be:

$$Adv1_{\mathcal{G}, \mathcal{A}(\lambda)} := |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$$

Assumption 2. Give a group generator (\mathcal{G}), we define the following distribution: $G = (N = p_1 p_2 p_3, G, G_T, e) \leftarrow_R \mathcal{G}$, $\alpha, s \leftarrow_R \mathbb{Z}_N$, $g \leftarrow_R G_{p_1}$, $g_2, X_2, Y_2 \leftarrow_R G_{p_2}$, $g_3 \leftarrow G_{p_3}$, $D = (G, g, g_2, g_3, g^\alpha X_2, g^s Y_2)$, $T_1 = e(g, g)^{\alpha s}$, $T_2 \leftarrow_R G_T$. We define the advantage of an algorithm (\mathcal{A}) in breaking Assumption 2 to be:

$$Adv2_{\mathcal{G}, \mathcal{A}(\lambda)} := |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$$

Assumption 3. Give a group generator (\mathcal{G}), we define the following distribution: $G = (N = p_1 p_2 p_3, G, G_T, e) \leftarrow_R \mathcal{G}$, $g, X_1 \leftarrow_R G_{p_1}$, $g_2 \leftarrow_R G_{p_2}$, $X_3 \leftarrow_R G_{p_3}$, $D = (G, g, g_2, X_1 X_3)$, $T_1 \leftarrow_R G_{p_1}$, $T_2 \leftarrow_R G_{p_1 p_3}$. We define the advantage of an algorithm (\mathcal{A}) in breaking Assumption 1 to be:

$$Adv3_{\mathcal{G}, \mathcal{A}(\lambda)} := |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$$

Assumption 4. Give a group generator (\mathcal{G}), we define the following distribution: $G = (N = p_1 p_2 p_3, G, G_T, e) \leftarrow_R \mathcal{G}$, $g, X_1 \leftarrow_R G_{p_1}$, $X_2, Y_2 \leftarrow_R G_{p_2}$, $g_3, Y_3 \leftarrow_R G_{p_3}$, $D = (G, g, g_3, X_1 X_2, Y_2 Y_3)$, $T_1 \leftarrow_R G$, $T_2 \leftarrow_R G_{p_1 p_3}$. We define the advantage of an algorithm (\mathcal{A}) in breaking Assumption 1 to be:

$$Adv4_{\mathcal{G}, \mathcal{A}(\lambda)} := |\Pr[\mathcal{A}(D, T_1) = 1] - \Pr[\mathcal{A}(D, T_2) = 1]|$$

2.4 System Model

The system model of this scheme is shown in Fig. 1. The system consists of three parts: control center (CC), district gateway (DGW) and user. It is assumed that the control center manages T areas, and each district gateway corresponds to DGW_1, \dots, DGW_T ; each area has n users, which are denoted as $\mathcal{L} = \{\text{ID}_1, \dots, \text{ID}_n\}$, and a detailed description of each entity is given below.

Control Center (CC). In this system model, the CC is an entity that is part of a district transport organization or an independent system operator. In the communication process, the CC is responsible for generating relevant system parameters, and finally sums up the power consumption information sent by the DGW.

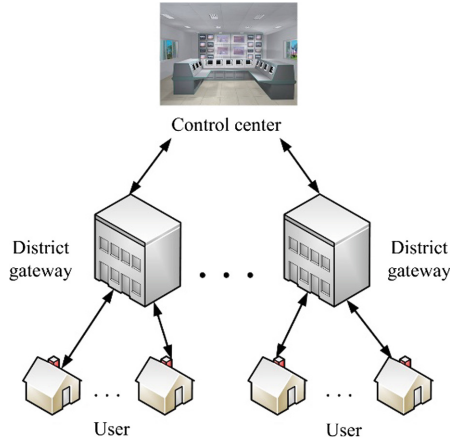


Fig. 1. System model

District Gateway (DGW). The DGW will receive signatures sent by the users $\mathcal{L} = \{\text{ID}_1, \dots, \text{ID}_n\}$ in the local district. After the verification, the total power consumption of users $\text{ID}_1, \dots, \text{ID}_n$ and the total power of each user are summarized.

User ($\text{ID}_1, \dots, \text{ID}_n$). Each user has a smart meter that collects the user's power usage information. The collected power information m is signed and sent to the DGW.

2.5 Security Model

A batch verification of linkable ring signature scheme is a tuple of probabilistic polynomial-time (PPT) algorithms below:

Setup. On input 1^λ where λ is a security parameter, the algorithm outputs master secret key α and system parameters $param$. The descriptions of user secret key space \mathcal{SK} , message space \mathcal{M} , identity space \mathcal{ID} as well as signature space \mathcal{SG} .

Extract. On input system parameter $param$, identity $\text{ID} \in \mathcal{ID}$ for a user and master secret key α , the algorithm outputs the user's secret key $SK_{\text{ID}} \in \mathcal{SK}$.

Sign. On input $param$, $\mathcal{L} = \{\text{ID}_1, \dots, \text{ID}_n\} \in \mathcal{ID}$, $m \in \mathcal{M}$, and secret key $\{SK_{\text{ID}_\pi} \in \mathcal{SK} | \text{ID}_\pi \in \mathcal{L}\}$, the algorithm outputs ring signature $\sigma \in \mathcal{SG}$.

Single Signature Verify. On input $param$, $\mathcal{L} = \{\text{ID}_1, \dots, \text{ID}_n\} \in \mathcal{ID}$, $m \in \mathcal{M}$, and signature $\sigma \in \mathcal{SG}$, the algorithm outputs Valid or Invalid.

Batch Verify. On input $param$, $\mathcal{L} = \{\text{ID}_1, \dots, \text{ID}_n\} \in \mathcal{ID}$, $m \in \mathcal{M}$, and η signatures $\sigma_1, \dots, \sigma_\eta \in \mathcal{SG}$, where $\mathcal{L}_1 = \dots = \mathcal{L}_\eta = \{\text{ID}_1, \dots, \text{ID}_n\}$, the algorithm outputs Valid or Invalid.

Link Verify. On input $param$, and signatures $\sigma_1, \sigma_2 \in \mathcal{SG}$, the algorithm outputs Link or Unlink.

3 Batch Verification Linkable Ring Signature (BVLRS)

We propose a batch verification linkable ring signature scheme (BVLRS) in the smart grid system. The scheme introduces batch verification [18, 22] and linkable ring signatures and consists of the following six algorithms:

Setup. CC chooses a bilinear group G of order $N = p_1 p_2 p_3$ (where p_1, p_2, p_3 are distinct primes). Let $H_0 : \{0, 1\}^* \rightarrow Z_N$, $H_1 : \{0, 1\}^* \times \{0, 1\}^* \rightarrow Z_N$ be two hash functions. Choose $g, h, u, v, w \in G_{p_1}$ and $\alpha \in Z_N$ as the master secret key. The public parameters are

$$param = \{N, g, h, u, v, w, e(g, g)^\alpha, H_0, H_1\}.$$

Extract. Each ring user generates their own private key, randomly generates $r, y, t_k \in {}_R Z_N$, compute $ID = H_0(ID)$ and $A = g^\alpha w^y, B = g^y, C = v^y (u^{ID} h)^r, D = g^r$. Output each user's private key $SK_{ID} = \{A, B, C, D, t_k\}$ about identity ID.

Sign. $\mathcal{L} = \{ID_1, \dots, ID_n\}$ is n users in the district gateway. We assume the user with identity ID_π is the actual signer, where $ID_\pi \in \mathcal{L}, \pi \in \{1, \dots, n\}$. To sign message $m \in \{0, 1\}^*$, compute $ID_i = H_0(ID_i)$ for $i = 1$ to n . Next compute $ID_{n+1} = H_1(m, \mathcal{L})$. Further, execute the following using private key $SK_{ID_\pi} = \{A, B, C, D, t_k\}$:

- (1) Randomly generate $x \in {}_R Z_N, y_i, r_i, \lambda_i \in {}_R Z_N$ for $i = 1$ to $n+1$ subject to the constraint that

$$\lambda_1 + \dots + \lambda_n + \lambda_{n+1} = 0 \quad (1)$$

- (2) When $i = 1$ to $n+1$, if $i \neq \pi$, then $A_i = g^{\lambda_i} w^{y_i}, B_i = g^{y_i}, C_i = v^{y_i} (u^{ID_i} h)^{r_i}, D_i = g^{r_i}$; if $i = \pi$, then $A_\pi = A g^{\lambda_\pi} w^{y_\pi}, B_\pi = B g^{y_\pi}, C_\pi = C v^{y_\pi} (u^{ID_\pi} h)^{r_\pi}, D_\pi = D g^{r_\pi}$.
- (3) Randomly generate $k \in {}_R Z_N$, compute $R = g^k, c = H_0(m, R), Q = k + t_k c \bmod N$.
- (4) Output signature $\sigma = \{[A_i, B_i, C_i, D_i]_{i=1}^{n+1}, R, Q, m, \mathcal{L}\}$.

Single Signature Verify. DGW receives the signature $\sigma = \{[A_i, B_i, C_i, D_i]_{i=1}^{n+1}, R, Q, m, \mathcal{L}\}$, first compute $ID_{n+1} = H_0(m, \mathcal{L})$ and $ID_i = H_0(ID_i)$, where $i = 1$ to n . Then randomly generate $s, t_1, \dots, t_{n+1} \in {}_R Z_N$ and check whether

$$\prod_{i=1}^{n+1} \frac{e(g^s, A_i) \cdot e(g^{t_i}, C_i)}{e(w^s v^{t_i}, B_i) \cdot e((u^{ID_i} h)^{t_i}, D_i)} \stackrel{?}{=} e(g, g)^{2s} \quad (2)$$

Output Valid if the equality holds. Otherwise output Invalid.

Batch Verify. DGW receives η signatures $\sigma_1, \dots, \sigma_\eta$, where $\mathcal{L}_1 = \dots = \mathcal{L}_\eta = \{ID_1, \dots, ID_n\}$. Same as above, compute $ID_i = H_0(ID_i)$, where $i = 1$ to $n+1$. Then randomly generate $s, t_1, \dots, t_{n+1} \in {}_R Z_N$, randomly select a smaller number T_1, \dots, T_η , and check whether

$$\prod_{i=1}^{n+1} \frac{e(g^s, \prod_{j=1}^{\eta} A_{ij}^{T_j}) \cdot e(g^{t_i}, \prod_{j=1}^{\eta} C_{ij}^{T_j})}{e(w^s v^{t_i}, \prod_{j=1}^{\eta} B_{ij}^{T_j}) \cdot e((u^{D_i} h)^{t_i}, \prod_{j=1}^{\eta} D_{ij}^{T_j})} \stackrel{?}{=} \prod_{j=1}^{\eta} e(g, g)^{z_s T_j} \quad (3)$$

Output Valid if the equality holds. Otherwise output Invalid.

Link Verify. DGW receives two signatures $\sigma_1 = \{R_1, Q_1, m_1, \cdot\}$, $\sigma_2 = \{R_2, Q_2, m_2, \cdot\}$ and public parameters $param$ compute:

$$tag_1 = (g^{Q_1} / R_1)^{\frac{1}{H_0(m_1, R_1)}} \quad (4)$$

$$tag_2 = (g^{Q_2} / R_2)^{\frac{1}{H_0(m_2, R_2)}} \quad (5)$$

Output Link if $tag_1 = tag_2$. Otherwise output Unlink.

4 Security Analysis

Theorem 1: BVLRS scheme is correct.

Proof: We substitute the private key SK_{ID} into the single signature verify, and can deduce:

$$\begin{aligned} & \prod_{i=1}^{n+1} \frac{e(g^s, A_i) \cdot e(g^{t_i}, C_i)}{e(w^s v^{t_i}, B_i) \cdot e((u^{D_i} h)^{t_i}, D_i)} \\ &= \frac{e(g^s, \prod_{i=1}^{n+1} A_i)}{e(w^s, \prod_{i=1}^{n+1} B_i)} \prod_{i=1}^{n+1} \left(\frac{e(g^{t_i}, C_i)}{e(v^{t_i}, B_i) \cdot e((u^{D_i} h)^{t_i}, D_i)} \right) \\ &= \frac{e(g^s, g^{z_s w^{y + \sum_{i=1}^{n+1} y_i}})}{e(w^s, g^{y + \sum_{i=1}^{n+1} y_i})} \prod_{i=1}^{n+1} \left(\frac{e(g, C_i)}{e(v, B_i) \cdot e((u^{D_i} h), D_i)} \right)^{t_i} \\ &= e(g, g)^{z_s} \left(\frac{e(g, C)}{e(v, B) \cdot e((u^{D_\pi} h), D)} \right)^{t_\pi} \\ &= e(g, g)^{z_s} \end{aligned}$$

Thus Eq. (2) holds, we then substituting private key SK_{ID} into Batch Verify:

$$\begin{aligned}
& \prod_{i=1}^{n+1} \frac{e(g^s, \prod_{j=1}^{\eta} A_{ij}^{T_j}) \cdot e(g^{t_i}, \prod_{j=1}^{\eta} C_{ij}^{T_j})}{e(w^s v^{t_i}, \prod_{j=1}^{\eta} B_{ij}^{T_j}) \cdot e((u^{D_i} h)^{t_i}, \prod_{j=1}^{\eta} D_{ij}^{T_j})} \\
&= \prod_{j=1}^{\eta} \prod_{i=1}^{n+1} \frac{e(g^s, A_{ij}^{T_j}) \cdot e(g^{t_i}, C_{ij}^{T_j})}{e(w^s v^{t_i}, B_{ij}^{T_j}) \cdot e((u^{D_i} h)^{t_i}, D_{ij}^{T_j})} \\
&= \prod_{j=1}^{\eta} e(g, g)^{\sum T_j}
\end{aligned}$$

Therefore, Eqs. (3) holds and BVLRS scheme is correct.

Theorem 2: BVLRS scheme is unforgeable under Assumptions 1, 2, 3 and 4.

Proof: We define the first part of the signature as $\sigma_{pt1} = \{[A_i, B_i, C_i, D_i]_{i=1}^{n+1}\}$ the latter part as $\sigma_{pt2} = \{R, Q\}$. The (m, \mathcal{L}) is public in the signature. The following is a proof process for σ_{pt1} unforgeability.

The first part of the signature has two types [23]: type-N and type-S. $\sigma_{pt1} = \{[A_i, B_i, C_i, D_i]_{i=1}^{n+1}\}$ generated by the signature algorithm is called type-N signature, and all components A_i, B_i are from group G_{p_1} only. σ_{pt1} is of type-S if it is not of type-N.

There are two types of keys: type-N and type-S. A key (A, B, C, D) generated by the key algorithm is called type-N key, and all components A, B are from group G_{p_1} only. A key is of type-S if it is not of type-N.

In proof part I, we show the adversary cannot output a forgery σ_{pt1} of type-S.

Proof: [Part I] Assume the adversary (\mathcal{A}) outputs a forgery σ_{pt1} of type-S, we show how to construct a simulator (\mathcal{S}) that breaks Assumption 1.

- *Setup.* \mathcal{S} constructs (g, X_3, T) . The parameter settings are as follows: \mathcal{S} randomly picks $\alpha, a, b, c, d \in_R \mathbb{Z}_N$, computes $h = g^a$, $u = g^b$, $v = g^c$, $w = g^d$, chooses two hash functions H_0, H_1 and gives $param = \{N, g, h, u, v, w, e(g, g)^\alpha, H_0, H_1\}$ to \mathcal{A} .
- *Query.* Since \mathcal{S} knows the master secret key (α) , \mathcal{S} can answer all the queries correctly.
- *Forgery.* \mathcal{A} outputs first part of the signature $\sigma_{pt1} = \{[A_i, B_i, C_i, D_i]_{i=1}^{n+1}\}$. \mathcal{S} computes $ID_{n+1} = H_0(m, \mathcal{L})$ and $ID_i = H_0(ID_i)$. We have, for any $s, t_1, \dots, t_{n+1} \in_R \mathbb{Z}_N$, that

$$\prod_{i=1}^{n+1} \frac{e(g^s, A_i) \cdot e(g^{t_i}, C_i)}{e(w^s v^{t_i}, B_i) \cdot e((u^{D_i} h)^{t_i}, D_i)} = e(g, g)^{\sum s}$$

- $\mathcal{A} \leftarrow$ outputs a type-S forgery, there exists A_j, B_j where j is an index. We show how it can be used to test if T contains a component in G_{p_2} . Without loss of generality, we assume either A_j or B_j contains an element in G_{p_2} .
- \mathcal{S} checks if

$$\prod_{i=1}^{n+1} \frac{e(T, A_i) \cdot e(g^{t_i}, C_i)}{e(T^{d \nu^{t_i}}, B_i) \cdot e((u^{ID_i} h)^{t_i}, D_i)} \stackrel{?}{=} e(g, T)^\alpha$$

If $T \in G_{p_1}$, there exists s such that $T = g^s$ and the above equation holds. Otherwise, there exists s, k such that $T = g^s g_2^k$ where g_2 is a generator of G_{p_2} . In that case the equation holds if and only if known $d \bmod p_2$. The probability of this happening will theoretically be negligible. Because all \mathcal{A} can infer that d is $d \bmod p_1$ which is unrelated to $d \bmod p_2$.

In proof part II, we show that the adversary cannot output a forgery $\sigma_{p_{t1}}$ of type-N. We use some games to prove the security of the scheme.

Game_{real}: This game is a real game and the key or signature returned to the adversary \mathcal{A} is of type-N.

Game_n: This game is a restricted game, that is, the ID of the adversary (\mathcal{A}) query and the challenge ID' cannot be equal to the modulo p_3 . At the same time, the hash value generated by \mathcal{A} is also distinguishable in modulo p_3 (i.e. \mathcal{A} cannot generate two ring identity sets and messages, $(m, \mathcal{L}) \neq (m', \mathcal{L}')$ but $H_1(m, \mathcal{L}) = H_1(m', \mathcal{L}')$).

Game_i: The first i queries of this game are answered is of type-S. Otherwise, the key or signature returned to the adversary (\mathcal{A}) is of type-N.

Game_k: We first show the behavior of the restricted adversary between *Game_{i-1}* and *Game_i* is the same for $i = 1$ to k where k is the number of queries made by the adversary. Finally, we show that probability of the adversary winning *Game_k* is negligible with another reduction.

Proof: [Part II] In this part of the proof, we assume the forgery $\sigma_{p_{t1}}$ of type-N.

We first show that \mathcal{A} is restricted in *Game_i* for $i = 1$ to k under Assumptions 3 and 4. Assume \mathcal{A} produces two values ID and ID' such that $ID \neq ID'$ and $ID = ID' \bmod p_3$. Let $P = \gcd(ID - ID', N)$. P is a non-trivial factor of N . In other words, $P \in (p_3, p_1 p_3, p_2 p_3)$. Let $q = N/P$. We consider the following two cases.

- (1) $(P, q) = (p_3, p_1 p_2) \vee (p_2 p_3, p_1)$. In this case we construct a simulator (\mathcal{S}) that breaks Assumption 3 as follows.

- *Setup*. \mathcal{S} constructs (g, g_2, X_1, X_2, T) . The parameter settings are as follows: \mathcal{S} randomly chooses $\alpha, a, b, c, d \in {}_R Z_N$, two hash functions H_0, H_1 and gives $param = \{N, g, h = g^a, u = g^b, v = g^c, w = g^d, e(g, g)^\alpha, H_0, H_1\}$ to \mathcal{A} .
- *Query*. When the j th query such that $j > i$, \mathcal{S} uses master secret key α to compute a key or signature $\sigma_{p_{t1}}$ of type-N. When $j \leq i$, \mathcal{S} computes a type-S key by randomly generating $y, r \in {}_R Z_N$ and $A = g^\alpha (X_1 X_3 g_2)^{dy}$, $B = (X_1 X_3 g_2)^y$, $C = (X_1 X_3 g_2)^{cy} (u^{ID} h)^r$, $D = g^r$. Likewise, a type-S signature can be

created by transforming a type-N signature σ_{p1} as follows using $y'_{n+1} \in {}_R Z_N$:
 $A_{n+1} := A_{n+1}(X_1 X_3 g_2)^{d y'_{n+1}}$, $B_{n+1} := B_{n+1}(X_1 X_3 g_2)^{y'_{n+1}}$, $C_{n+1} := C_{n+1}(X_1 X_3 g_2)^{c y'_{n+1}}$, $D_{n+1} := D_{n+1}$. Note that only components A_{n+1} , B_{n+1} , C_{n+1} contain elements in G_{p_2} and G_{p_3} .

- *Output.* \mathcal{S} first checks if it is the case that $(P, q) = (p_3, p_1 p_2) \vee (p_2 p_3, p_1)$ via testing if $g^q = 1$ and $(X_1 X_3)^q \neq 1$. This check ensures that $p_1 | q$ and $p_3 \nmid q$. Finally, \mathcal{S} tests whether $T^q = 1$ or not and break Assumption 3.

(2) $(P, q) = (p_1 p_3, p_2)$. In this case we construct a simulator (\mathcal{S}) that breaks Assumption 4 as follows.

\mathcal{S} constructs $(g, g_3, X_1 X_2, Y_2 Y_3, T)$. The rest of steps are the same as Setup in 1), \mathcal{S} gives a *param* to \mathcal{A} . Simultaneously, \mathcal{S} keeps random value $\psi \in {}_R Z_N$ secret.

When $j \leq i$, \mathcal{S} computes a type-S key and signature by randomly generating $y, r, y'_{n+1} \in {}_R Z_N$ and $A = g^z w^y (Y_2 Y_3)^{\psi y}$, $B = (g Y_2 Y_3)^y$, $C = v^y (Y_2 Y_3)^{c y} (u^{ID} h)^r$, $D = g^r$, $A_{n+1} := A_{n+1}(Y_2 Y_3)^{\psi y'_{n+1}}$, $B_{n+1} := B_{n+1}(Y_2 Y_3)^{y'_{n+1}}$, $C_{n+1} := C_{n+1}(Y_2 Y_3)^{c y'_{n+1}}$, $D_{n+1} := D_{n+1}$. Note that only components A_{n+1} , B_{n+1} , C_{n+1} contain elements in G_{p_2} and G_{p_3} .

\mathcal{S} first checks if it is the case that $(P, q) = (p_1 p_3, p_2)$ via testing if $g^q \neq 1$ and $g_3^q \neq 1$. This check ensures that $p_1 | q$ and $p_3 \nmid q$. Since $Pq = N$ and both P and q are non-trivial factors of N , it implies $P = p_1 p_3$. \mathcal{S} tests whether $T^P = 1$ or not and break Assumption 4.

As a second step of the proof, we need to show that the behavior of a restricted adversary in $Game_{i-1}$ and $Game_i$ is the same for $i = 1$ to k . We first can use two oracles \mathcal{O}_0 and \mathcal{O}_1 [21]. Finally, we show how to construct a simulator (\mathcal{S}) that distinguishes oracle \mathcal{O}_0 and \mathcal{O}_1 , thus either breaking Assumption 3 or 4.

Furthermore, we present a reduction of \mathcal{A} that produces a type-N forgery in $Game_k$ to simulator \mathcal{S} that breaks Assumption 2.

- *Setup.* \mathcal{S} constructs $(g, g_2, g_3, g^z X_2, g^s Y_2, T)$ and its task is to determine whether $T = e(g, g)^{zs}$ or not. The parameter settings are as follows: \mathcal{S} randomly picks $a, b, c, d \in {}_R Z_N$, $h = g^b$, $u = g^a$, $v = g^c$, $w = g^d \in {}_R G_{p_1}$, two hash functions H_0, H_1 and gives *param* = $\{N, g, h, u, v, w, e(g, g^z X_2), H_0, H_1\}$ to \mathcal{A} . Note that \mathcal{S} does not know master secret key α .
- *Extract Query.* To answer an extract query on identity ID such that $ID = H_0(ID)$, \mathcal{S} chooses $y, r, f \in {}_R Z_N$ and computes

$$\begin{aligned} A &= (g^z X_2)^{d+1} w^y (g_2 g_3)^{f(d+1)}, \\ B &= g^z X_2 g^y (g_2 g_3)^f, \\ C &= (g^z X_2)^c v^y (u^{ID} h)^r (g_2 g_3)^{fc}, \\ D &= g^r. \end{aligned}$$

- *Signature Query.* To answer a signature query on ring $\mathcal{L}=\{\text{ID}_1, \dots, \text{ID}_n\}$ on message m such that $\text{ID}_i = H_0(\text{ID}_i)$ for $i = 1$ to n and $\text{ID}_{n+1} = H_1(m, \mathcal{L})$. \mathcal{S} chooses $f, \lambda_1, r_1, y_1, \dots, \lambda_n, y_n, r_n, \lambda'_{n+1}, y_{n+1}, r_{n+1} \in_R \mathcal{Z}_N$, where $\lambda_1 + \dots + \lambda_n + \lambda'_{n+1} = 0$, and computes for $i = 1$ to n :
 $A_i = g^{\lambda_i} w^{y_i}, B_i = g^{y_i}, C_i = v^{y_i} (u^{\text{ID}_i} h) r_i, D = g^{r_i}$.

\mathcal{S} then computes the following:

$$\begin{aligned} A_{n+1} &= (g^x X_2)^{d+1} g^{\lambda'_{n+1}} w^{y'_{n+1}} (g_2 g_3)^{f(d+1)}, \\ B_{n+1} &= g^x X_2 g^{y'_{n+1}} (g_2 g_3)^f, \\ C_{n+1} &= (g^x X_2)^c v^{y'_{n+1}} (u^{\text{ID}_{n+1}} h)^r (g_2 g_3)^{fc}, \\ D_{n+1} &= g^{r_{n+1}}. \end{aligned}$$

- *Forgery.* \mathcal{A} outputs a $\sigma_{pr1} = \{[A_i, B_i, C_i, D_i]_{i=1}^{n+1}\}$ on message m and ring $\mathcal{L}=\{\text{ID}_1, \dots, \text{ID}_n\}$. \mathcal{S} first computes $\text{ID}_{n+1} = H_1(m, \mathcal{L})$ and $\text{ID}_i = H_0(\text{ID}_i)$. We have, for any $s, t_1, \dots, t_{n+1} \in_R \mathcal{Z}_N$, that

$$\prod_{i=1}^{n+1} \frac{e(g^s, A_i) \cdot e(g^{t_i}, C_i)}{e(w^s v^{t_i}, B_i) \cdot e((u^{\text{ID}_i} h)^{t_i}, D_i)} = e(g, g)^{zs}$$

- \mathcal{S} picks $t_1, \dots, t_{n+1} \in_R \mathcal{Z}_N$ and computes

$$e(g, g)^{zs} := \prod_{i=1}^{n+1} \frac{e(g^s Y_2, A_i) \cdot e(g^{t_i}, C_i)}{e((g^s Y_2)^d v^{t_i}, B_i) \cdot e((u^{\text{ID}_i} h)^{t_i}, D_i)}$$

\mathcal{S} can test whether $T = e(g, g)^{zs}$ or not and breaks Assumption 2.

Finally, we prove that $\sigma_{pr2} = \{R, Q\}$ is unforgeable.

Proof: Assume adversary \mathcal{A} is successful, i.e., (m^*, R^*, Q^*) is a valid forgery. By definition we then have $(g^{Q^*}/R^*)^{1/H_0(m^*, R^*)} = \text{tag} = g^{tk}$, i.e., $g^{Q^*} = R^* (g^{tk})^{H_0(m^*, R^*)}$, hence in particular (R^*, Q^*) is a valid Schnorr signature on m^* . We know that the Schnorr signature scheme is strongly existentially unforgeable if *DLP* [20] is hard in G and H is modeled as a random oracle. Therefore, $\sigma_{pr2} = \{R, Q\}$ is unforgeable.

Theorem 3: BVLRS scheme is anonymous

Proof: The challenger constructs the master key as well as private keys SK_{ID_0} and SK_{ID_1} for identities ID_0 and ID_1 following the setup and extract algorithms.

For any challenge signature $\sigma = \{[A_i, B_i, C_i, D_i]_{i=1}^{n+1}, R, Q, m, \mathcal{L}\}$ created using SK_{ID_b} on message m and ring \mathcal{L} , there exists values $\mathcal{R}_0 := \{(\lambda_{i,0}, y_{i,0}, r_{i,0})_{i=1}^{n+1}, k_0\}$ and $\mathcal{R}_1 := \{(\lambda_{i,1}, y_{i,1}, r_{i,1})_{i=1}^{n+1}, k_1\}$ such that σ is created from private key SK_{ID_0} using randomness \mathcal{R}_0 or SK_{ID_1} using randomness \mathcal{R}_1 . We know \mathcal{R}_0 and \mathcal{R}_1 have identical

distributions. Therefore, even computationally unbounded adversary cannot distinguish the actual signer with probability better than random guessing.

Theorem 4: BVLRS scheme is linkable

Proof: We know from *Theorem 2* that non-ring members cannot forge valid ring signatures. We only need to consider ring members to forge signatures linked with known signatures. Assuming that the known ring signature is δ , \mathcal{A} forged a ring signature linked with the δ is δ' . Because \mathcal{A} is a ring member, \mathcal{A} only knows his private key. Same as the proof method in *Theorem 2*, if \mathcal{A} can forge δ' without knowing the private key corresponding to δ . This is the same as solving the *DLP* hard problem. Hence, our scheme is linkable.

5 Performance Analysis

First, we analyze the computational complexity of the BVLRS scheme. We define according to the literature [24]: An exponentiation computation by E , a bilinear pairing computation by P , a hash function computation by H_f , n is the number of ring members and η is the number of signatures. The computational complexity analysis of the BVLRS scheme is shown in Table 1.

Table 1. Complexity analysis of BVLRS

Algorithm	Computation cost
Setup	$1P$
Extract	$7E$
Sign	$[7(n+1) + 1]E$
Single Signature Verify	$(4n+7)E + (4n+4)P$
Batch Verify	$[4n\eta + 4n + 4\eta + 6]E + [4n + \eta + 4]P$
Link Verify	$2E$

Table 1 can clearly see the computational complexity of these six algorithms. Next, we compare the computational efficiency with the BVLRS scheme and other ring signature schemes under the standard model. Comparison of computational efficiency is shown in Table 2.

Table 2. Comparison of computational efficiency

Ring signature	Single signature verification computations	η signatures batch verification computations
Ref. [18]	$(4n+7)E + (4n+4)P$	$(4n\eta + 7\eta)E + (4n\eta + 4\eta)P$
Ref. [25]	$E + (4n+2)P + nH_f$	$(4n+2)\eta P + \eta E + n\eta H_f$
Ref. [26]	$E + (2n+4)P + nH_f$	$(2n+4)\eta P + \eta E + n\eta H_f$
BVLRS	$(4n+7)E + (4n+4)P$	$[4n\eta + 4n + 4\eta + 6]E + [4n + \eta + 4]P$

According to Table 2, the computational efficiency of the BVLRS scheme is greatly improved in batch verification. We quoted the literature [24], knowing that the time of a exponentiation operation is 0.58 ms, the time of a bilinear pairing operation is 10.31 ms, and the time of hash function operation is 3.58 ms. The batch verification computational complexity of the literature [18, 25, 26] and BVLRS is shown in Figs. 2, 3, 4 and 5.

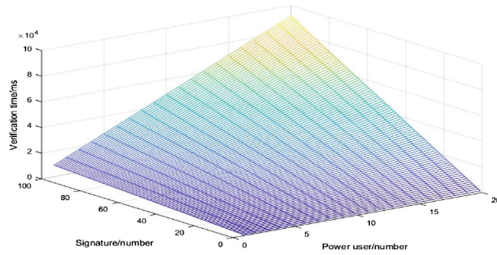


Fig. 2. Ref. [18]

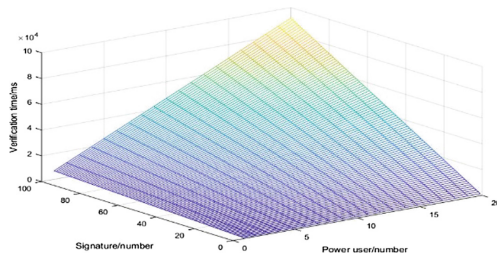


Fig. 3. Ref. [25]

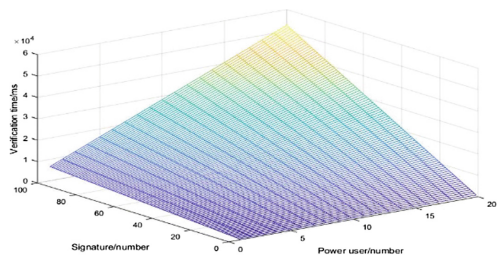


Fig. 4. Ref. [26]

As can be seen from Figs. 2, 3, 4, and 5, we have greatly reduced the computation time for ring signature batch verification. Finally, we analyze the security features of the BVLRS scheme and compare it with other schemes, as shown in Table 3.

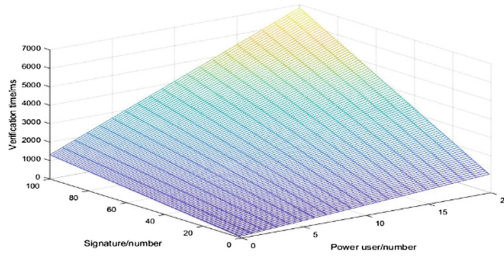


Fig. 5. BVLRS

Table 3. Comparison of security features.

Ring signature	Anonymity	Unforgeability	Linkability
Ref. [18]	√	√	×
Ref. [25]	√	√	×
Ref. [26]	√	√	×
BVLRS	√	√	√

According to Table 3, we know that the BVLRS scheme is linkable. This feature guarantees that the DGW is calculating the total power consumption of each user without knowing the specific identity of the user, and can also determine malicious electricity users based on this feature.

6 Conclusion

In this paper, BVLRS scheme is proposed under the standard model, which can be used in the secure communication of smart grid. Through security and performance analysis, we know that the BVLRS scheme is anonymous, unforgeable, and linkable. The linkable feature guarantees that the DGW is calculating the total power consumption of each user without knowing the specific identity of the user, and can also determine malicious electricity users based on this feature. We compare the BVLRS scheme with other ring signatures under the standard model. The BVLRS scheme can significantly reduce the computational cost of ring signature verification when the selected \mathcal{L} is the same.

References

1. Fang, X., Misra, S., Xue, G., Yang, D.: Smart grid - the new and improved power grid: a survey. *IEEE Commun. Surv. Tutorials* **14**(4), 944–980 (2012)
2. Anderson, R., Fuloria, S.: Who controls the off switch. In: *IEEE International Conference on Smart Grid Communications*, pp. 96–101 (2010)

3. Roberto, D.P., Pietro, M., Refik, M.: Confidentiality and integrity for data aggregation in WSN using peer monitoring. *Wireless Personal Commun.* **2**(2), 181–194 (2009)
4. Lu, R., Liang, X., Li, X., Lin, X., Shen, X.: EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications. *IEEE Trans. Parallel Distrib. Syst.* **23**(9), 1621–1631 (2012)
5. Ruj, S., Nayak, A.: A decentralized security framework for data aggregation and access control in smart grids. *IEEE Trans. Smart Grid* **4**(1), 196–205 (2013)
6. Erkin, Z.: Private data aggregation with groups for smart grids in a dynamic setting using CRT. In: *IEEE International Workshop on Information Forensics & Security*, pp. 1–6 (2015)
7. Shen, H., Zhang, M., Shen, J.: Efficient privacy-preserving cube-data aggregation scheme for smart grids. *IEEE Trans. Inf. Forensics Secur.* **12**(6), 1369–1381 (2017)
8. Abdallah, A.R., Shen, X.S.: A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid. *IEEE Trans. Smart Grid* **9**(1), 396–405 (2018)
9. Lang, B., Wang, J., Cao, Z.: Multidimensional data tight aggregation and fine-grained access control in smart grid. *J. Inf. Sec. Appl.* **40**, 156–165 (2018)
10. Gope, P., Sikdar, B.: Lightweight and privacy-friendly spatial data aggregation for secure power supply and demand management in smart grids. *IEEE Trans. Inf. Forensics Secur.* **14**(6), 1554–1566 (2019)
11. Cheung, J.C.L., Chim, T.W., Yiu, S.M.: Credential-based privacy-preserving power request scheme for smart grid network. In: *Global Telecommunications Conference*, pp. 1–5 (2011)
12. Yu, C.-M., Chen, C.-Y., Kuo, S.-Y.: Privacy-preserving power request in smart grid networks. *IEEE Syst. J.* **8**(2), 441–449 (2014)
13. Badra, M., Zeadally, S.: Design and performance analysis of a virtual ring architecture for smart grid privacy. *IEEE Trans. Inf. Forensics Secur.* **9**(2), 321–329 (2014)
14. Tan, X., Zheng, J., Zou, C., Niu, Y.: Pseudonym-based privacy-preserving scheme for data collection in smart grid. *IJAHUC* **22**(2), 120–127 (2016)
15. 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)
16. Guan, Z., et al.: Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities. *IEEE Commun. Mag.* **56**(7), 82–88 (2018)
17. Li, S., Choi, K., Chae, K.: OCPM: ortho code privacy mechanism in smart grid using ring communication architecture. *Ad Hoc Netw.* **22**(16), 93–108 (2014)
18. Au, M.H., Liu, J.K., Susilo, W., Zhou, J.: Realizing fully secure unrestricted ID-based ring signature in the standard model based on HIBE. *IEEE Trans. Inf. Forensics Secur.* **8**(12), 1909–1922 (2013)
19. Günther, F., Poettering, B.: Linkable message tagging: solving the key distribution problem of signature schemes. In: *ACISP*, pp. 195–212 (2015)
20. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on Ciphertexts. In: *TCC*, pp. 325–341 (2005)
21. Lewko, A.B., Waters, B.: Unbounded HIBE and attribute-based encryption. In: *EUROCRYPT*, pp. 547–567 (2011)
22. Ferrara, A.L., Green, M., Hohenberger, S., Pedersen, M.S.: Practical short signature batch verification. In: *CT-RSA*, pp. 309–324 (2009)
23. Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: *TCC*, pp. 455–479 (2010)

24. Ming, Y., Zhang, X., Shen, X.: Efficient privacy-preserving multi-dimensional data aggregation scheme in smart grid. *IEEE Access* **7**, 32907–32921 (2019)
25. Malavolta, G., Schröder, D.: Efficient ring signatures in the standard model. *ASIACRYPT* **2**, 128–157 (2017)
26. Ren, H., Zhang, P., Shentu, Q., Liu, J.K., Yuen, T.H.: Compact ring signature in the standard model for blockchain. In: *ISPEC*, pp. 50–65 (2018)



Hierarchical Identity-Based Signature over Verifiable Random Function

Juan Ren^(✉) and Leyou Zhang

School of Mathematics and Statistics, Xidian University, Xi'an 710126, China
juaner_r@126.com

Abstract. Hierarchical computation makes an important role in constructing identity-based signature (IBS) since it provides a delegation mechanism to IBS, which results in the Hierarchical identity-based signature (HIBS). HIBS has widely potential applications in the large networks. However, the constructions available cannot propose a good trade-off for the private keys and signatures since the size of private keys or signatures depends on the identity depth. In this paper, a new hierarchical computation algorithm is introduced to construct HIBS scheme. The new scheme achieves $O(1)$ -size private keys and signatures, which are independent of identity depth. It is the best trade-off at present. Furthermore, under the $n + 1 - weak$ Computational Diffie-Hellman Exponent ($n + 1 - wCDH$) assumption, the scheme is provably secure against existential forgery in the standard model.

Keywords: Hierarchical computation · Verifiable random function · IBS · Constant size private keys · Standard model · Provable security

1 Introduction

In 1984, Shamir showed the public key could be easily issued using given a receiver's identity, which is called identity-based encryption (IBE) [1]. IBE supports a sender to encrypt a message by the user's identity as a public key. The first practical IBE appeared in 2001, which was presented by Boneh and Franklin [2] based on pairing, where the security was achieved under the random oracle model. The first construction without random oracles was proposed in [3]. IBE has the desirable advantages over other public key encryption (PKE) schemes, such as a single Private Key Generator (PKG) would completely eliminate online lookup. However, the above is also a bottleneck for a large network because the single PKG must complete all the procedures, such as the identity verification, generalization of the private keys and transmitting them in a secure channel. How to overcome them is a challenge problem in IBE. Gentry et al. [4, 5] gave a

This work was supported in part by the National Cryptography Development Fund under Grant (MMJJ20180209), International S&T Cooperation Program of Shaanxi Province No. 2019KW-056.

good method by using the hierarchical delegation computation, which issues the hierarchical IBE (HIBE). In an HIBE, the root PKG distributes the workload to lower-level PKGs: a parent PKG (root PKG) needs only to generate private keys for its offspring, who in turn generate private keys for their offspring (domains) in the next level. All procedures such as Authentication and private key transmission can be done locally. The first practical HIBE in the standard model was due to Boneh and Boyen [3]. More works are proposed since then, such as [6–9]. The most recent works are constructed based on hard problems over lattices [10, 11] without using pairings. Chow et al. [11] proposed a new hierarchical ID-based signature that shared the same system parameters with their hierarchical ID-based encryption scheme.

Hierarchical identity-based signature (HIBS) is a natural extension application of hierarchical delegation computation in IBE. In 2015, Wang et al. [23] introduced Key Privacy Authorities (KPA) to restrict the power of PKG. However, PKG-KPA models bring significant overhead because of the extra identity authentication and the more complicated key generating algorithms. Many constructions [12–14, 22, 23] were proposed now. However, these constructions have undesirable features such as constructing in a weak model, security relying on the random oracle model and the signatures and private keys size depending on identity levels. The short signature is useful for applications. Au et al. [15] issued an escrow-free IBS model that each signer used a public key and a secret key to sign messages. But in their schemes, a judge and a Trusted Third Party are required in their model. Zhang et al. [16, 17, 22] proposed an escrow-free IBS scheme that unnecessarily depends on any judges. However some new shortcomings appeared: the public keys size in [16] was too large and the security in [17] is reduced to a strong hardness assumption. Abdalla et al. [18] proposed a methodology to construct verifiable random functions (VRF) from a class of identity based key encapsulation mechanisms (IB-KEM) that called VRF suitable. Wu [19] also proposed an efficient scheme but it was a designated identity-based signature and not an efficient HIBS. In addition, these schemes did not solve the trade-off between the private keys and signature since the size of private keys or signature depended on the hierarchy depth. Recent constructions based on lattice also have the above limits [20, 21]. Recently, Li et al. [26] proposed the formal definition and security model of identity-based broadcast encryption (IBBE) and constructed an IBBE scheme with continuous leakage resilience. Furthermore, Some secure cryptographic primitives [24, 25] are given in the length-bounded leakage model.

In this paper, we take aims at the construction of the efficient and practical HIBS. Then a new HIBE is proposed. The new scheme inherits the desirable advantages in the previous works, such as constructed without random oracles and short signatures. The main contributions of our works are exciting since the new scheme achieves short public keys and $O(1)$ -size private keys. They are important and useful for a large scale network. Finally, the new work is also provably secure under the $n + 1 - wCDH$ assumption.

This paper is organized as follows. In Sect. 2, some definitions are given. The new works appears in Sect. 3. Security analysis is introduced in Sect. 4. Finally, we conclude this paper.

2 Preliminaries

2.1 Bilinear Pairing

Let G and G_1 be two (multiplicative) cyclic groups of prime order p and g be a generator of G . A bilinear map e is a map $e : G \times G \rightarrow G_1$ with the following properties:

- Bilinearity: for all $u, v \in G$, $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$;
- Non-degeneracy: $e(g, g) \neq 1$;
- Computability: there is an efficient algorithm to compute $e(u, v)$ for all $u, v \in G$.

2.2 Hardness Assumption

Security of our scheme will be reduced to the hardness of the $n + 1 - wCDH$ problem. We briefly recall the definition of the $n + 1 - CDH$ problem at first.

Definition 2.1 (Computational Diffie-Hellman Exponent Problem: $n + 1 - CDH$).

Given a group G of prime order p with generator g and elements $g^a, g^{a^2}, \dots, g^{a^n} \in G$ where a is selected uniformly at random from \mathbb{Z}_p and $n \geq 1$, the $n + 1 - CDH$ problem in G is to compute $g^{a^{n+1}}$.

In this paper, a weak version is used which is called $n + 1 - wCDH$. It is given as follows.

Definition 2.2 (weak Computational Diffie-Hellman Exponent Problem: $n + 1 - wCDH$). Given a group G of prime order p with generator g and elements $g^a, g^{a^2}, \dots, g^{a^n}, g^{a^{n+2}}, g^{a^{2n}} \in G$ where a is selected uniformly at random from \mathbb{Z}_p and $n \geq 1$, the $n + 1 - wCDH$ problem in G is to compute $g^{a^{n+1}}$.

It comes from the $n + 1$ weak Computational Bilinear Diffie-Hellman Exponent Problem which is introduced in [8]. Another description can be found in [17].

Definition 2.3. We say that the (t, ϵ) $n + 1 - wCDH$ assumption holds in a group G , if no adversary running in time at most t can solve the $n + 1 - wCDH$ problem in G with probability at least ϵ .

2.3 L-Level HIBS Scheme

We propose an l -level HIBS scheme consisting of four algorithms *Setup*, *Extract*, *Sign* and *Verify*. They are specified as follows:

Setup: Given a security parameter, *PKG* returns the system parameters together with the master key. The system parameters are publicly known while the master key is known only to the *PKG*.

Key generation and Delegation: Given an identity $ID = (v_1, \dots, v_k)$, the public parameters and the private key d_{ID} corresponding to the identity $ID_{k-1} = (v_1, \dots, v_{k-1})$, it returns a private key d_{ID} for ID . The identity ID is used as the public key while d_{ID} is the corresponding private key.

Sign: Given the identity ID , the private key and a message M from the message space, it outputs a signature σ corresponding to the M and ID .

Verify: Given the signature corresponding to the M and ID , it is accepted if $Verify(PK, M, \sigma) = \text{“Valid”}$. Otherwise it is rejected.

2.4 Existential Unforgeability

The general security definition for a signature scheme is called existential unforgeability under a chosen identity and message attack. It works using the following game between a challenger and an adversary \mathbf{A} .

Setup: The challenger runs algorithm *KeyGen* to obtain a public key PK . The adversary \mathbf{A} is given PK .

Queries: Proceeding adaptively, \mathbf{A} requests queries for private keys and signatures as follows.

- *Extract:* \mathbf{A} chooses an identity ID and gives to challenger. The challenger computes $d_{ID} = Extract(ID)$ and returns d_{ID} to \mathbf{A} .
- *Sign:* \mathbf{A} chooses an identity ID , a message M and sends them to challenger. The challenger computes $\sigma = Sign(d_{ID}, M)$ and returns σ to \mathbf{A} .

Forgery: The adversary \mathbf{A} outputs a pair (M^*, ID^*, σ^*) and wins the game if

- ID^* and any prefix of ID^* does not appear in any *Extract* query. Moreover, any *Sign* query on (M^*, ID') , where ID' is ID^* or any prefix of ID^* , does not appear in the Queries phase too.
- $Verify(PK, M^*, \sigma^*) = Valid$.

We will use a weaker notion of security which is called existential unforgeability under adaptively chosen message and selective identity attack. Here we require that the adversary submits a challenge identity before seeing the public key. This notion is defined using the following game between a challenger and an adversary \mathbf{A} .

Init: The adversary \mathbf{A} outputs an identity ID^* (the selective ID) that he wants to attack.

Setup: The challenger runs algorithm *KeyGen* to obtain a public key PK . The adversary \mathbf{A} is given PK .

Queries: Proceeding adaptively, \mathbf{A} issues the queries of *Extract* and *sign* as follows.

- *Extract*: \mathbf{A} chooses an identity ID ($ID \neq ID^*$ or prefix of ID^*) and gives to challenger. The challenger computes $d_{ID} = \text{Extract}(ID)$ and returns d_{ID} to \mathbf{A} .
- *Sign*: \mathbf{A} chooses an identity ID ($ID \neq ID^*$ or prefix of ID^*), a message M and gives to challenger. The challenger computes $\sigma = \text{Sign}(d_{ID}, M)$ and sends it to \mathbf{A} .

Forgery: The adversary \mathbf{A} outputs a tuple (M^*, ID^*, σ^*) and wins the game if

- ID^* and any prefix of ID^* does not appear in any *Extract* query. Moreover, any *Sign* query on (M^*, ID') , where ID' is ID^* or any prefix of ID^* , does not appear in the Queries phase too.
- $\text{Verify}(PK, M^*, \sigma^*) = \text{Valid}$.

Definition 2.4. An adversary \mathbf{A} is called a (t, q_e, q_s, ϵ) forger of an HIBS scheme if \mathbf{A} has the advantage at least ϵ in the above game after running the game in time at most t , making at most q_e and q_s extract queries signature queries, respectively.

A signature scheme is (t, q_e, q_s, ϵ) secure if no (t, q_e, q_s, ϵ) forger exists.

2.5 Our Approach-Verifiable Random Function

Recently, Abdalla, Catalano and Fiore [18] proposed a direct method to build a verifiable random functions (VRF) suitable IB-KEM scheme. It did not need to resort to the inefficient Goldreich Levin transform. In their construction, a useful function called ACF-“Hash function” was introduced. We found it is also suitable for HIBS to achieve constant size private keys. It works as follows.

Input: Pick $\alpha, \alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n$ in Z_p at random. And choose g randomly in G .

Output: Let $ID = (v_1, \dots, v_n)$ be a n -bit string representing an identity, where $v_i \in \{0, 1\}$. Let $h_0 = g$, then for $i = 1, \dots, n$, compute $h_i = (h_{i-1})^{\alpha_i^{v_i} \beta_i^{1-v_i}}$. Finally, output

$$h_n = (h_{n-1})^{\alpha_n^{v_n} \beta_n^{1-v_n}} = g^{\prod_{i=1}^n \alpha_i^{v_i} \beta_i^{1-v_i}}.$$

3 New Construction

3.1 Our Works

Following [7, 9, 17], an identity is a bit-string of length n . The new HIBS is constructed as follows.

Setup: Let G denote a group with a prime order p . Pick $\alpha, \alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n$ in Z_p at random for $1 \leq i \leq l$ (l is the maximum depth of HIBS). Select a random generator g of G and set $g_1 = g^\alpha$. Then choose $g_2, u_0, u_1, \dots, u_t$ randomly in G . The public key is

$$PK = \{g, g_1, g_2, u_0, u_1, \dots, u_t\}.$$

The master key is g_2^α . At hierarchy depth i , PKG_i is given the sharing keys

$$Msk_i = \{\alpha_{i1}, \dots, \alpha_{in}, \beta_{i1}, \dots, \beta_{in}\}.$$

Key Generation-Hierarchical Computing:

- *Root private keys generation:* For the first level $ID = (v_1)$ with $v_1 = (v_{11}, \dots, v_{1n})$ and $v_{1i} \in \{0, 1\}$, root PKG computes the auxiliary parameters at first as follows: Let $h_{10} = g$. PKG computes $T(v_1) = \prod_{i=1}^n \alpha_{1i}^{v_{1i}} \beta_{1i}^{1-v_{1i}}$ and sets $h_{1n} = g^{T(v_1)}$. Then it computes the private key for users as follows.

$$d_{ID} = (d_0, d_1) = (g_2^\alpha h_{1n}^r, g^r)$$

where $r \in Z_p$.

- *Delegation Hierarchical computation:* For the k -th level $ID = (v_1, \dots, v_k)$ ($k \leq l$) with $v_i = (v_{i1}, \dots, v_{in})$ and $v_{ij} \in \{0, 1\}$, by using the parent $(k-1)$ -th level $ID = (v_1, \dots, v_{k-1})$ and the corresponding private key

$$d'_{ID} = (d'_0, d'_1) = (g_2^\alpha (\prod_{i=1}^{k-1} h_{in})^r, g^r).$$

PKG_k first generates the auxiliary information parameters as follows: Let $h'_{k0} = d'_1$. PKG_k computes

$$h'_{kn} = (d'_1)^{T(v_k)} = (g^r)^{\prod_{j=1}^n \alpha_{kj}^{v_{kj}} \beta_{kj}^{1-v_{kj}}}.$$

Let $h_{kn} = g^{\prod_{j=1}^n \alpha_{kj}^{v_{kj}} \beta_{kj}^{1-v_{kj}}}$. Then one can obtain $h'_{kn} = (h_{kn})^r$. The private key for ID is generated as

$$d_{ID} = (d_0, d_1) = (d'_0 h'_{kn}, d'_1) = (g_2^\alpha (\prod_{i=1}^k h_{in})^r, g^r).$$

Note: h_{in} is outputted as pubic key for $1 \leq i \leq k$.

Sign: Let $M = (m_1, \dots, m_t)$ be a message to be signed and $m_i \in \{0, 1\}$. Choose a random $s \in Z_p^*$ and generate the signature for M in the following manner:

$$\sigma = (\sigma_1, \sigma_2, \sigma_3) = (d_0 (u_0 \prod_{i=1}^t u_i^{m_i})^s, d_1, g^s)$$

Verify: After receiving the signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$, the verifier will verify the following equation holds or not.

$$e(\sigma_1, g) = e(g_1, g_2) e(\prod_{i=1}^k h_{in}, \sigma_2) e(u_0 \prod_{i=1}^t u_i^{m_i}, \sigma_3).$$

If it is true, the signature is accepted. Otherwise it will be rejected.

3.2 Correctness

If it is a valid signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ of M , then

$$\begin{aligned}
 e(\sigma_1, g) &= e(d_0(u_0 \prod_{i=1}^t u_i^{m_i})^s, g) \\
 &= e(g_2^\alpha (\prod_{i=1}^k h_{in})^r (u_0 \prod_{i=1}^t u_i^{m_i})^s, g) \\
 &= e(g_2^\alpha, g) e((\prod_{i=1}^k h_{in})^r, g) e((u_0 \prod_{i=1}^t u_i^{m_i})^s, g) \\
 &= e(g_1, g_2) e(\prod_{i=1}^k h_{in}, \sigma_2) e(u_0 \prod_{i=1}^t u_i^{m_i}, \sigma_3)
 \end{aligned}$$

3.3 A Distinct Feature

In the previous construction, each potential PKG_i has to be sent master keys. It results in an additional transmission cost for root PKG . However, it also generates an efficient verification algorithm for private keys. It works as follows:

New-Setup: Pick $\alpha, \alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_n$ in Z_p at random for $1 \leq i \leq l$. Set $g_1 = g^\alpha$, $t_{ij} = g^{\alpha_{ij}}$, $T_{ij} = g^{\beta_{ij}}$ for $1 \leq i \leq l, 1 \leq j \leq n$, then choose $g_2, u_0, u_1, \dots, u_t$ randomly in G . The public key is

$$PK = \{g, g_1, g_2, u_0, u_1, \dots, u_t, t_{ij}, T_{ij}\}_{1 \leq i \leq l, 1 \leq j \leq n}.$$

The master key is g_2^α . At hierarchy depth i , PKG_i is given the sharing keys

$$Msk_i = \{\alpha_{i1}, \dots, \alpha_{in}, \beta_{i1}, \dots, \beta_{in}\}.$$

New-Key Generation-Hierarchical Computing:

- For the first level $ID = (v_1)$ with $v_1 = (v_{11}, \dots, v_{1n})$ and $v_{1i} \in \{0, 1\}$, root PKG can generate the auxiliary information parameters as follows: Let $h_{10} = g$. For $1 \leq i \leq n$, PKG computes $h_{1i} = (h_{1(i-1)})^{\alpha_{1i}^{v_{1i}} \beta_{1i}^{1-v_{1i}}}$. Then the private key for ID is generated as follows:

$$d_{ID} = (d_0, d_1) = (g_2^\alpha h_{1n}^r, g^r)$$

where $r \in Z_p$. Then h_{11}, \dots, h_{1n} are outputted as PK .

- For the k -th level with $ID = (v_1, \dots, v_k)$ ($k \leq l$) with $v_i = (v_{i1}, \dots, v_{in})$ and $v_{ij} \in \{0, 1\}$, by using the parent $(k-1)$ -th level $ID = (v_1, \dots, v_{k-1})$ and the corresponding private key

$$d'_{ID} = (d'_0, d'_1) = (g_2^\alpha (\prod_{i=1}^{k-1} h_{in})^r, g^r).$$

PKG_k first generates the auxiliary information parameters as follows: Let $h'_{k0} = d'_1$. For $1 \leq i \leq n$, computes $h'_{ki} = (h'_{k(i-1)})^{\alpha_{ki}^{v_{ki}} \beta_{ki}^{1-v_{ki}}}$. Then $h'_{kn} = (g^r)^{\prod_{j=1}^n \alpha_{kj}^{v_{kj}} \beta_{kj}^{1-v_{kj}}}$. Let $h_{kn} = g^{\prod_{i=1}^n \alpha_{ki}^{v_{ki}} \beta_{ki}^{1-v_{ki}}}$. Then one can obtain $h'_{kn} = (h_{kn})^r$. The private key for ID is generated as

$$d_{ID} = (d_0, d_1) = (d'_0 h'_{kn}, d'_1) = (g_2^\alpha (\prod_{i=1}^k h_{in})^r, g^r).$$

Note: All h_{ij} is outputted as pubic key for $1 \leq i \leq k, 1 \leq j \leq n$.

For each user can verify the validity as follows: If $v_{ij} = 1$, checking $e(g, h_{ij}) = e(t_{ij}, h_{i(j-1)})$; otherwise, checking $e(g, h_{ij}) = e(T_{ij}, h_{i(j-1)})$. If it is valid, he/she can check

$$e(d_0, g) = e(g_1, g_2) = e(\prod_{i=1}^k h_{in}, d_1).$$

4 Security Analysis and Efficiency

4.1 Security

In this section, we give the security analysis as follows.

Theorem 4.1. If the $n + 1 - wCDH$ assumption holds in G , then the new scheme is secure.

Proof: Suppose there exists a selective CPA adversary \mathbf{A} that is able to win the above game with advantage ϵ , then an algorithm \mathbf{B} can be constructed to solve the $n + 1 - wCDH$ problem with advantage $\frac{\epsilon}{2^{kn}}$. Suppose \mathbf{B} has been given a tuple

$$(g^a, g^{a^2}, \dots, g^{a^n}, g^{a^{n+2}}, g^{a^{2n}}).$$

The game works as follows:

Init: \mathbf{A} first declares the identity $ID^* = (v_1^*, \dots, v_k^*)$ with $k \leq l$ that it wants to attack, where $v_i^* = (v_{i1}^*, \dots, v_{ik}^*)$.

Setup: \mathbf{B} first generates the system parameters for \mathbf{A} . It sets $g_1 = y_1$ and selects randomly

$$\gamma, \alpha_{i,j}, \beta_{i,j}, v_0, \dots, v_t \in Z_p^*.$$

where $1 \leq i \leq l, 1 \leq j \leq n, 0 \leq j \leq t$. It sets

$$g_2 = y_n g^\gamma = g^{a_n + \gamma}.$$

The master key is g_2^α . For any level i , the master keys MSk_i are set as

$$MSk_i = \{\alpha_{i,j} \alpha^{v_{i,j}^*}, \beta_{i,j} \alpha^{1-v_{i,j}^*}\}.$$

The public key is

$$PK = \{g, g_1, g_2, \dots, u_0, \dots, u_t\}$$

Finally, **B** sends the **PK** to **A**. The corresponding master keys are unknown to **B**.

Queries: **A** will generate a series of queries and **B** returns the corresponding answers in the following way:

- *Extract queries:* **A** generates up to q_e extract queries. Each of them q_i is given as follows. Let $ID = (v_1, \dots, v_k)$ denote the corresponding identity. It is worth noting that $ID \neq ID^*$ or its prefix. This restriction denote that there is a j such that $v_j \neq v_j^*$. To give the answers, **B** first computes the followings:

$$h_{i1} = \begin{cases} g_{i1}^{\alpha_{i1}^{v_{i1}} \beta_{i1}^{1-v_{i1}}} & \text{if } v_{i1} \neq v_{i1}^* \\ y_1^{\alpha_{i1}^{v_{i1}} \beta_{i1}^{1-v_{i1}}} & \text{if } v_{i1} = v_{i1}^* \end{cases} \quad (1)$$

$$h_{i2} = \begin{cases} y_1^{\alpha_{i2}^{v_{i2}} \beta_{i2}^{1-v_{i2}}} & \text{if } v_{i2} \neq v_{i2}^* \\ y_1^{\alpha_{i2}^{v_{i2}} \beta_{i2}^{1-v_{i2}}} & \text{if } v_{i2} = v_{i2}^* \wedge v_{i1} \neq v_{i1}^* \\ y_2^{\alpha_{i2}^{v_{i2}} \beta_{i2}^{1-v_{i2}}} & \text{if } v_{i2} = v_{i2}^* \wedge v_{i1} = v_{i1}^* \end{cases} \quad (2)$$

where $1 \leq i \leq k$. Then all parameters can be obtained. In order to generate the private keys for ID , **B** first computes the private keys for $ID_j = (v_1, \dots, v_j)$ where j denotes the first element such that $v_j \neq v_j^*$. Without loss of generality, we set t to denote the number of positions such that $v_{j,i} = v_{j,i}^*$. Then we can obtain

$$h_{in} = y_n^{T(v_1)}, \dots, h_{(j-1)i} = y_n^{T(v_{j-1})}, h_{jn} = y_t^{T(v_j)},$$

where $T(v_k) = \prod_{i=1}^n \alpha_{ki}^{v_{ki}} \beta_{ki}^{1-v_{ki}}$ for $1 \leq k \leq j$ and $t < n$.

B picks randomly $r' \in Z_p$ and generates the private key as follows:

$$d_{ID} = (d_0, d_1) = (g_2^\alpha \left(\prod_{i=1}^j h_{in} \right)^r, g^r),$$

where $r = r' - \frac{\alpha^{n-t+1}}{T(v_j)}$.

In fact, one can verify the following equation holds.

$$\begin{aligned} g_2^\alpha \left(\prod_{i=1}^j h_{in} \right)^r &= y_{n+1} y_1^\gamma \left(\prod_{i=1}^j h_{in} \right)^{r' - \frac{\alpha^{n-t+1}}{T(v_j)}} \\ &= y_1^\gamma \left(\prod_{i=1}^{j-1} y_n^{T(v_i)} \right)^{r'} \left(\prod_{i=1}^{j-1} y_{2n-t+1}^{-\frac{T(v_i)}{T(v_j)}} (y_t^{T(v_j)}) \right)^{r'} \end{aligned}$$

Since y_{n+1} disappeared, the rest of elements are known to **B**. Thus, **B** can simulate the first element of the private keys. The second element, g^r can be set as $y_{n-t+1}^{-\frac{1}{T(v_j)}} g^{r'}$ (since $0 < t < n$, y_{n-t+1} is known to **B**). So **B** can complete the private keys simulation. Next **B** can generate the private keys of $ID = (v_1, \dots, v_k)$ using the private keys of $ID_j = (v_1, \dots, v_j)$.

- *Signing Queries:* **A** will issue a signature query for $M = (m_1, \dots, m_t)$ under user's identity $ID = (v_1, \dots, v_k)$. It first makes an extraction query on ID using the above manner. Then **B** will construct a signature using the above private keys as follows:

$$\sigma = (\sigma_1, \sigma_2, \sigma_3) = (g_2^\alpha (\prod_{i=1}^k h_{in})^r (g^{\sum_{k=1}^t v_k^{m_i} + v_0})^s, g^r, g^s)$$

In fact,

$$\begin{aligned} \sigma &= (\sigma_1, \sigma_2, \sigma_3) \\ &= (g_2^\alpha (\prod_{i=1}^k h_{in})^r (g^{\sum_{k=1}^t v_k^{m_i} + v_0})^s, g^r, g^s) \\ &= (d_0 (u_0 \prod_{i=1}^t u_i^{m_i})^s, d_1, g^s) \end{aligned}$$

Forgery: **A** outputs an identity $ID^* = (v_1^*, \dots, v_j^*)$ and message M^* . **B** computes the auxiliary parameters for challenge identity ID^* as follows.

$$h_{in} = g^{T(v_1^*)}, \dots, h_{(k-1)n} = g^{T(v_{k-1}^*)}, \dots, h_{kn} = g^{T(v_k^*)}$$

Then **B** sends the corresponding the private keys to **A**. If **A** can break this scheme: a signature of M^* for ID^* is valid. Then **B** can solve the $n+1$ - w CDH problem. In fact, let $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ denote the forged signature. Then

$$\begin{aligned} \sigma^* &= (\sigma_1^*, \sigma_2^*, \sigma_3^*) \\ &= (g_2^\alpha (\prod_{i=1}^k h_{in})^r (g^{\sum_{k=1}^t v_k^{m_i^*} + v_0})^s, g^r, g^s) \\ &= (g_2^\alpha \prod_{i=1}^k (g^{T(v_i^*)})^r (g^{\sum_{k=1}^t v_k^{m_i^*} + v_0})^s, g^r, g^s) \\ &= (y_{n+1} y_1^y \prod_{i=1}^k (g^{T(v_i^*)})^r (g^{\sum_{k=1}^t v_k^{m_i^*} + v_0})^s, g^r, g^s). \end{aligned}$$

Hence

$$\frac{\sigma_1^*}{y_1^\gamma \prod_{i=1}^k (\sigma_2^*)^{T(v_i^*)} (\sigma_2^*)^{\sum_{k=1}^t v_k^{m_i^*} + v_0}} = y_{n+1}.$$

This shows that **B** has solved the $n+1$ - w CDH problem. Note: from the received elements, **A** gets nothing on the ID^* chosen by **B**, thus such a choice is achieved with probability $\frac{1}{2^{kn}}$.

4.2 Efficiency Analysis

From Sect. 3.1, we use the ACF-“Hash function” $h_{kn} = g^{\prod_{i=1}^n \alpha_{ki}^{v_{ki}} \beta_{ki}^{1-v_{ki}}}$ in the delegation algorithm to shrink the size of the private keys. Observe that for

Table 1. Comparison of security

Scheme	Hardness assumption	OR	Security model
[11]	CDH	YES	s -ID
[12]	CDH	YES	Gs -ID
[15]	q -SDH	NO	FULL
[16]	n -CDH	NO	FULL
[17]	q -SDH	NO	FULL
[20]	SIS	NO	s -ID
[21]	SIS	NO	s -ID
Ours	$n + 1 - w$ CDH	NO	s -ID

identities at any depth, the private keys contain only 2 group elements. The signatures also are made up of 3 elements. They are the admiring features over the available since both private keys and signatures achieve $O(1)$ -size, which solve the trade-off between the private keys size and signatures size. In addition, the public keys in our scheme achieve $k + t + 4$, which is a shorter size comparing the existing HIBS schemes. Though no pairs are used in [20, 21], the trade-off between private keys and signatures is not solved. The cost of verifying algorithm in our scheme needs 3 pairing operations (the value $e(g_1, g_2)$ can be precomputed). Tables 1 and 2 compare our proposed scheme with other HIBS schemes.

In Tables 1 and 2, OR denotes the random oracles. PK and pk are the public key and private key respectively. Full and s -ID denote the adaptive and selective-identity security respectively. In addition, k and l are the user's hierarchy depth and the maximum hierarchy depth of the scheme. SIS is "short integer solution" problem [20]. In addition, in Table 2, $m, n, \lambda_1, \lambda_2$ denote security parameters and $m_1 = O(ln), m_2 = O(\lambda_1 ln)$.

Table 2. Comparison of computational efficiency

Scheme	PK size	pk size	Signature size	Pairing
[11]	$(l + 2) G $	$(k + 1) G $	$(k + 2) G $	3
[12]	$(l + 2) G $	$(k + 1) G $	$(k + 2) G $	$k + 2$
[15]	$[2l + 1] G $	$(l - t + 2) G $	$2 G + p$	4
[16]	$[(n + 1)l + t + 3] G $	$[(n + 1)(l - t) + 2] G $	$3 G $	4
[17]	$(l + t + 4) G $	$(l - k + 3) G $	$5 G + p$	4
[20]	$(1 + 2l\lambda_1 m_2 + 2\lambda_2)nm_1 + n$	$(m_1 + k\lambda_1 m_1)^2$	$(1 + k\lambda_1 + \lambda_2)m_1 + n$	0
[21]	$(n + 2l\lambda_1 m_2 + \lambda_2 n)m_2 + n$	m_2^2	$2m_2 + n$	0
Ours	$(k + t + 4) G $	$2 G $	$3 G $	3

Compare with [12], the computational cost of our scheme is shorter, which the pairing operation is constant in our scheme. The results in Table 1 reveal that

our scheme is more flexible and secure. As shown in Table 2, our scheme achieves short public keys and $O(1)$ -size private keys, which it is obviously superior to other schemes [11, 12, 15–17, 20, 21] in efficiency.

5 Conclusion

To overcome the shortcomings in the existing HIBS schemes, a new HIBS scheme is introduced. The new scheme has short public parameters and achieves $O(1)$ – size signatures. Furthermore, it also achieves constant size private keys, which is independent of identity scale. Short signatures and constant size private keys are important for a large scale network since they reduced the storage and computation cost to users. Under the $n + 1 - w$ CDH assumption, the propose scheme is provably secure.

Unfortunately, the security of our scheme is reduced to the $n + 1 - w$ CDH problem—a strong assumption. It is always an open problem to construct a more efficient scheme with strong security and based on a natural assumption. In addition, we hope the future works also shrink the public keys size.

References

1. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Advance in Cryptography, pp. 47–53. ACM, Santa Barbara (1984)
2. Boneh, D., Franklin, M.: Identity based encryption from the Weil pairing. SIAM J. Comput. **32**(3), 586–615 (2001)
3. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_14
4. Gentry, C.: Practical identity-based encryption without random oracles. In: 24th Annual International Conference on The Theory and Applications of Cryptographic Techniques, pp. 445–464. ACM, Saint Petersburg (2006)
5. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-36178-2_34
6. Waters, B.: Dual key encryption: realizing fully secure IBE and HIBE under simple assumption. In: 29th Annual International Cryptology Conference on Advances in Cryptology, pp. 619–636. ACM, Santa Barbara (2009)
7. Zhang, L., Hu, Y., Wu, Q.: Hierarchical Identity-Based Encryption with Constant size private keys. ETRI J. **34**(1), 142–145 (2012)
8. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_26
9. Cash, D., Hofheinz, D., Kiltz, E.: Bonsai trees, or how to delegate a lattice basis. In: 29th Annual International Conference on Theory and Applications of Cryptographic Techniques, pp. 523–552. ACM, French Riviera (2010)
10. 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

11. Chow, S.S.M., Hui, L.C.K., Yiu, S.M., Chow, K.P.: Secure hierarchical identity based signature and its application. In: Lopez, J., Qing, S., Okamoto, E. (eds.) ICICS 2004. LNCS, vol. 3269, pp. 480–494. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30191-2_37
12. Li, J., Zhang, F., Wang, Y.: A new hierarchical ID-based cryptosystem and CCA-secure PKE. In: Zhou, X., et al. (eds.) EUC 2006. LNCS, vol. 4097, pp. 362–371. Springer, Heidelberg (2006). https://doi.org/10.1007/11807964_37
13. Au, M., Liu, J., Yuen, T., et al.: Practical Hierarchical Identity Based Encryption and Signature schemes Without Random Oracles. Cryptology ePrint Archive, Report 2006/308 (2006)
14. Yuen, T., Susilo, W., Mu, Y.: How to construct identity-based signatures without the key escrow problem. *Int. J. Inf. Secur.* **9**(4), 297–311 (2010)
15. Au, M., Liu, J., Yuen, T., et al.: Efficient Hierarchical Identity Based Signature in the Standard Model. Cryptology ePrint Archive, Report 2007/68 (2007)
16. Zhang, L., Hu, Y., Wu, Q.: New construction of short hierarchical ID-based signature in the standard model. *Fundamenta Informaticae* **90**(1–2), 191–201 (2009)
17. Zhang, L., Hu, Y., Wu, Q.: Adaptively secure hierarchical identity-based signature in the standard model. *J. China Univ. Posts Telecommun.* **17**(6), 95–100 (2010)
18. Abdalla, M., Catalano, D., Fiore, D.: Verifiable random functions from identity-based key encapsulation. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 554–571. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_32
19. Wu, Q., Zhang, L.: New efficient hierarchical identity-based signature. *J. Comput.* **8**(3), 803–810 (2013)
20. Rückert, M.: Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles. In: Sendrier, N. (ed.) PQCrypto 2010. LNCS, vol. 6061, pp. 182–200. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12929-2_14
21. Tian, M., Huang, L., Yang, W.: A new hierarchical identity-based signature scheme from lattices in the standard model. *Int. J. Netw. Secur.* **14**(6), 310–315 (2012)
22. Zhang, X., Xu, C., Jin, C., Xie, R.: Efficient forward secure identity-based shorter signature from lattice. *Comput. Electr. Eng.* **40**(6), 1963–1971 (2014)
23. Wang, X., Chen, P., Zhou, H., Su, J.: T-HIBE: a trustworthy and secure hierarchical identity-based encryption system. *Chin. J. Electron* (2015)
24. Li, J., Guo, Y., Yu, Q., Lu, Y., Zhang, Y.: Provably secure identity-based encryption resilient to post-challenge continuous auxiliary inputs leakage. *Secur. Commun. Netw.* **9**(10), 1016–1024 (2016)
25. Li, J., Teng, M., Zhang, Y., Yu, Q.: A leakage-resilient CCA-secure identity-based encryption scheme. *Comput. J.* **59**(7), 1066–1075 (2017)
26. Li, J., Yu, Q., Zhang, Y.: Identity-based broadcast encryption with continuous leakage resilience. *Inf. Sci.* **429**(3), 177–193 (2018)

Attack and Behavior Detection



Analysis of Ciphertext Policy Hidden Attribute-Based Encryption and Its Improved Method

Gongcheng Hu^(✉) and Leyou Zhang

School of Mathematics and Statistics, Xidian University, Xi'an 710126, China
gchenghu@126.com

Abstract. With people paying more attention to personal privacy protection, how to achieve fine-grained access control of data while protecting users' privacy has become a hot research at present. A Ciphertext Policy Attribute-based Encryption (CP-ABE) with hiding policy is regarded as one of the most effective methods to solve above problem. Although many policy hidden CP-ABE schemes have been proposed, in this paper, we will show some of them fail to achieve the complete privacy-preserving. Hence two effective attacks are introduced at first, namely, the attack of attribute testing and the guessing attack of access policy. Then we show several known schemes can not resist these two attacks. Finally, an effective policy hidden CP-ABE scheme that can resist the above two attacks is proposed. And we also show it achieves full security in the standard model under static assumptions.

Keywords: Policy hidden · Privacy protection · Ciphertext policy attribute-based encryption

1 Introduction

Since Waters proposed Attribute-based encryption (ABE) mechanism in [1], public key encryption has changed from traditional “one-to-one” to “one-to-many”, which greatly improves the efficiency of data sharing. With the rapid development of cloud computing and Internet of Things, ABE has been applied to many fields of them. In order to meet the needs of different fields, different types of ABE are proposed by researchers, such as Ciphertext policy attribute-based encryption (CP-ABE) [2], Key policy attribute-based encryption (KP-ABE) [3] and Double policy attribute-based encryption (DP-ABE) [4]. In a CP-ABE mechanism, secret key is related to the user's attribute list, while the ciphertext is associated with an access structure defined by the encryptor. Therefore, a data owner in CP-ABE system has full access control over his/her data, namely, whether a data user can access these data depending on whether he/she is authorized

Supported by the National Cryptography Development Fund of China under Grant (MMJJ20180209).

by the data owner. Due to this characteristic of CP-ABE, it is widely applied in the fields of smart medical cloud, smart grid, smart home and so on [5,6]. However, in these fields, there are a large number of user privacy security issues happened. Especially, people have paid more attention to the protection of personal privacy after the Snowden leaks. Considering such an example shown in Fig. 1: in a smart medical cloud system, A data user defines such a policy $\langle\langle\text{“D hospital” AND “TB expert”}\rangle\rangle$ OR $\langle\langle\text{“F hospital” AND “Cardiologist”}\rangle\rangle$ to encrypt his personal health information and uploads the encrypted information data to a medical cloud. It is obvious that this access structure contains two sensitive attributes, namely, *“TB expert”* and *“Cardiologist”*. If there is a malicious data user who wants to know the health status of the data owner, he can judge that the data owner suffers from heart disease or tuberculosis only by the access policy. Therefore, it is necessary to hide access policy in a CP-ABE mechanism in terms of protecting user privacy.

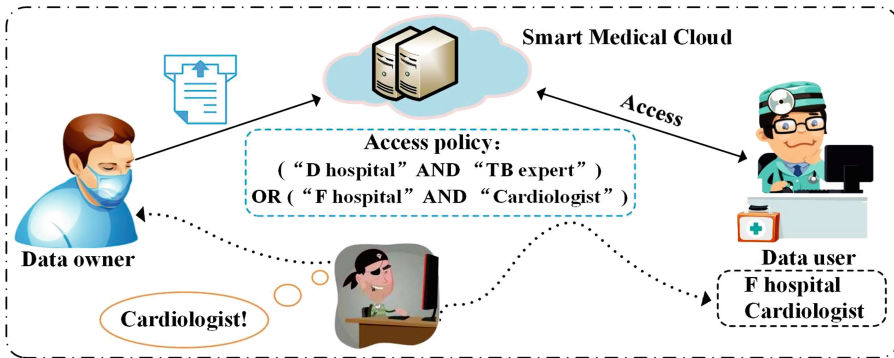


Fig. 1. An example of privacy leakage in Smart Medical Cloud

1.1 Related Work

In a CP-ABE algorithm, hiding access policy is considered as one of the most effective methods to avoid user privacy leakage. The idea of policy hidden in CP-ABE was first proposed by Nishide *et al.* [7] and an access policy in their scheme was not sent with the ciphertext, but embedded implicitly in the ciphertext. In order to further hide access policy, each attribute in their scheme has multiple candidate values, which means that the candidate value of an attribute is different, and the access policy composed of this attribute is also different. Subsequently, Malluhi *et al.* [8] proposed another algorithm based on these and the size of ciphertext in the scheme does not linearly increase with the number of attributes, which greatly saves the storage overhead of ciphertext. Due to the access policy embedded in the ciphertext, a decryptor has to do a lot of decryption calculations to determine whether he/she is an authorized user, which is infeasible for the system with limited computing resources. To deal

with this problem, Zhang *et al.* [9] proposed a scheme to support decryption testing. In the decryption phase of their scheme, a decryptor needs to carry out a test operation first that consumes a small amount of computing resources. If the test stage is passed, the decryption operation will be executed, otherwise, the decryptor is unauthorized and the decryption calculation will be terminated. In addition, some other optimization schemes are also proposed to improve the efficiency and practicability of hiding policy CP-ABE algorithm [10–12].

However, all the above algorithms are constructed based on the restricted AND-gate. In order to improve the flexibility of access structure, some schemes based on flexible access control LSSS have been proposed. Waters [13] first proposed the ABE algorithm based on LSSS. It is worth mentioning that the access policy of this scheme can be any type of boolean expression, but a decryptor in the decryption phase must obtain the access matrix M associated with the access policy and the map ρ between the associated M and decryption user's attribute. It is difficult to hide the access policy in the schemes based on LSSS because the access matrix M and the map ρ have to be sent along with the ciphertext. To address the problem, Lai [14] proposed a scheme based on attribute hierarchy, that is, an attribute in their scheme is composed of two parts: attribute name and attribute value. In their algorithm, the attribute name is published, while the attribute value is confidential and embedded in the ciphertext, which realizes partial hiding of the access policy. Subsequently, a scheme with fully hiding access policy was proposed by Khan *et al.* [15]. Their idea is that both the access matrix M and the map ρ were not sent along with the ciphertext, but the encryptor calculated the constant set $\mathcal{W} = \{\omega_i\}_{i \in [1, l]}$ used for decryption and embedded the set \mathcal{W} in the ciphertext. Although this method can hide access structure, it consumes a lot of computing resources of an encryptor in the encryption phase. Another scheme with fully hiding access policy was proposed by Yang *et al.* [16]. In this scheme, the function of M and ρ were replaced by a Bloom filter, namely, the index set for ciphertext and secret key pairing can be calculated by the Bloom filter in the decryption phase. However, the false recognition rate is natural characteristic of a bloom filter, that is, when the false recognition rate exceeds the threshold value, the wrong result will be output by their decryption algorithm. Recently, Zhang *et al.* [17] proposed a scheme with fast decryption, where only constant bilinear pairing operation is required in the decryption stage. What's more, some other optimization algorithms have also been proposed [18–20].

1.2 Our Contribution

Although many policy hidden algorithms have been proposed, but we find that some of them cannot actualize privacy-preserving well. In a hidden policy CP-ABE algorithm, some ciphertext components $C_{i,j}$ are associated with attribute in access policy, which is also called the access policy embedded in ciphertext. Some random elements r_i are chosen from \mathbb{Z}_p in most existing schemes to blind these components $C_{i,j}$ to achieve access policy hidden. In order to decrypt successfully, the ciphertext has to contain the component $C_j = g^{r_i}$,

which also leads to obvious correlation between $C_{i,j}$ and C_j . In this paper, we propose two effective attack methods based on these characteristics: (1) the attack of attribute testing, (2) the guessing attack of access policy. In addition, we also propose an effective policy hidden CP-ABE scheme and its attribute correlation values $C_{i,j}$ and the blind factor correlation components C_j will be blinded twice, which can eliminate the correlation between them and resist the above two attacks. And the proposed scheme supports outsourcing decryption to reduce the computational overhead of a decryptor.

2 Preliminaries

In this part, some basic cryptographic definitions and symbolic descriptions in the paper will be given in detail.

2.1 Composite Order Bilinear Maps

Let \mathcal{G} be an algorithm which takes a security parameter 1^λ as input and output a tuple $(G, G_T, e, p_1, p_2, p_3)$, where p_1, p_2, p_3 are three different primes, G and G_T are cyclic groups with order $N = p_1 p_2 p_3$. If $e : G \times G \rightarrow G_T$ is a bilinear map, it has the following three characteristics:

1. (*Bilinearity*) $\forall w, f \in G, \alpha, \beta \in \mathbb{Z}_N, e(w^\alpha, f^\beta) = e(w, f)^{\alpha\beta}$.
2. (*Computable*) $\forall w, f \in G$, then $e(w, f)$ is efficiently computable.
3. (*Non-degenerate*) $\exists w \in G$ such that $e(w, w)$ has order N in G_T . Let $G_{p_1}, G_{p_2}, G_{p_3}$ denote that subgroups of G , respectively. It also emphasize that if $g_1 \in G_{p_1}, g_2 \in G_{p_2}, e(g_1, g_2) = 1$. In fact, $g_{p_j}, (j = 1, 2, 3)$ be the generator of G_{p_j} , respectively. Therefore, $\forall \alpha_j \in \mathbb{Z}_N$, then $e(g_{p_j}^{\alpha_j}, g_k^{\alpha_k}) = 1, (j \neq k \wedge j, k = 1, 2, 3)$.

2.2 Prime Order Bilinear Map

Let \mathcal{Q} be an generate algorithm which inputs security parameter 1^k and outputs a tuple (G, G_T, e, p) , where G and G_T are cyclic groups with order p , e is a bilinear map which maps a element of G to $G_T (e : G \times G \rightarrow G_T)$. In addition, it also chooses two generators g and f of the group G . Generally speaking, a prime order bilinear map has three characteristics:

1. (*Bilinearity*) $\forall g, f \in G, a, c \in \mathbb{Z}_p^*, e(g^a, f^c) = e(g, f)^{ac}$.
2. (*Computable*) $\forall g, f \in G$, then $e(g, f)$ is efficiently computable.
3. (*Non-degenerate*) $\exists g \in G$ such that $e(g, g) \neq 1$.

2.3 Access Structure

It's hard to hide access policy in traditional LSSS-based ABE schemes because its access matrix M and the map ρ are both sent along with the ciphertext. In order to address this problem, our idea is to disperse attribute function, namely, an attribute consists of an attribute value and its name. Attribute value contains important information and embedded in ciphertext, while attribute names can be disclosed because the information it contains is irrelevant. We define $\mathbb{M} = (M, \rho, \mathcal{T})$ as an access policy in the proposed scheme, where $\mathcal{T} = (\tau_{\rho(1)}, \tau_{\rho(2)}, \dots, \tau_{\rho(m)})$ ($\tau_{\rho(x)} \in \Psi_{\rho(x)}$) is the attribute value set in \mathbb{M} . Suppose that an user whose attribute list $L = (N_L, V_L)$ satisfies the access policy \mathbb{M} , and the equation $\tau_{\rho(x)} = \varphi_{\rho(x)}$ is valid for $x \in \Gamma$, where $\Gamma = \{x | \rho(x) \in N_L\}$, $N_L \subset \{1, 2, \dots, n\}$ denotes the user's attribute name index set and $V_L = \{\phi_1, \phi_2, \dots, \phi_m\}$ ($\phi_i \in \Psi_i$) is its attribute value set. A linear secret sharing schemes includes two sub-algorithms.

Secret sharing: Let Φ be the attribute universe, which consists of n types of attributes, and $\Phi = (att_1, att_2, \dots, att_n)$. Each attribute $att_i \in \Phi$ has n_i values and Ψ_i is a set that consists of all possible values for attribute att_i , where $\Psi_i = \{\varphi_1, \varphi_2, \dots, \varphi_{n_i}\}$. It also sets M is a $m \times n$ access matrix over \mathbb{Z}_p and ρ is a map from each row of M to an attribute name index. (i.e., $\rho: \{1, 2, \dots, m\} \rightarrow \{1, 2, 3, \dots, n\}$). When a secret value $s \in \mathbb{Z}_p$ is shared by this algorithm, it first sets a vector $\vec{v} = (s, z_2, z_3, \dots, z_n)$ and calculates $\mu_x = M_x \cdot \vec{v}$, where $\{z_2, z_3, \dots, z_n\} \in_R \mathbb{Z}_p$, $\{\mu_x\}_{x \in m}$ are the m shared values of s and M_x denotes the x^{th} row of M .

Secret reconstruction: Let P be an index set of attribute name for authorized user, then there exists a constant set $\mathbb{C} = \{\omega_x\}_{x \in \Gamma'}$ such that $\sum_{x \in \Gamma'} \omega_x M_x = (1, 0, \dots, 0)$, and the secret value s can be reconstructed by $\sum_{x \in \Gamma'} \mu_x \omega_x$ where $\Gamma' = \{x | \rho(x) \in P\}$.

2.4 Complexity Assumptions

The group order in our scheme is a composite number, which is the product of three different primes. We now state the complexity assumptions used in the proposed scheme.

Assumption 1. Let \mathcal{G} be the algorithm mentioned above and define a distribution tuple $\mathcal{D} = (\Omega, g_1, g_3)$, then the advantage of \mathcal{A} in breaking the assumption is defined as:

$$Adv_{\mathcal{A}}^1 = |\Pr[\mathcal{A}(\mathcal{D}, \mathcal{B}_0) = 1] - \Pr[\mathcal{A}(\mathcal{D}, \mathcal{B}_1) = 1]|$$

where $\Omega = (N, p_1, p_2, p_3, G, G_T, e)$, $g_1 \in G_{p_1}$, $g_2 \in G_{p_2}$, $\mathcal{B}_0 \in G$ and $\mathcal{B}_1 \in G_{p_1 p_2}$.

Theorem 1. If the algorithm \mathcal{G} satisfies Assumption 1, for any polynomial time adversary \mathcal{A} , its advantage $Adv_{\mathcal{A}}^1$ is negligible.

Assumption 2. Let \mathcal{G} be the algorithm mentioned above and define a distribution tuple $\mathcal{D} = (\Omega, g_1, \mathcal{X}_2, \mathcal{X}_1\mathcal{X}_3, \mathcal{Y}_1\mathcal{Y}_3)$, then the advantage of \mathcal{A} in breaking the assumption is defined as:

$$Adv_{\mathcal{A}}^2 = |\Pr[\mathcal{A}(\mathcal{D}, \mathcal{B}_0) = 1] - \Pr[\mathcal{A}(\mathcal{D}, \mathcal{B}_1) = 1]|$$

where $\Omega = (N, p_1, p_2, p_3, G, G_T, e)$, $g_1, \mathcal{X}_1, \mathcal{Y}_1 \in G_{p_1}$, $\mathcal{X}_2 \in G_{p_2}$, $\mathcal{X}_3, \mathcal{Y}_3 \in G_{p_3}$, $\mathcal{B}_0 \in G_{p_1 p_3}$ and $\mathcal{B}_1 \in G_{p_1}$.

Theorem 2. *If the algorithm \mathcal{G} satisfies Assumption 2, for any polynomial time adversary \mathcal{A} , its advantage $Adv_{\mathcal{A}}^2$ is negligible.*

Assumption 3. Let \mathcal{G} be the algorithm mentioned above and define a distribution tuple $\mathcal{D} = (\Omega, g_1, g_1^\alpha \mathcal{X}_2, \mathcal{X}_3, g_1^s \mathcal{Y}_1 \mathcal{Y}_2, \mathcal{Z}_2)$, then the advantage of \mathcal{A} in breaking the assumption is defined as:

$$Adv_{\mathcal{A}}^3 = |\Pr[\mathcal{A}(\mathcal{D}, \mathcal{B}_0) = 1] - \Pr[\mathcal{A}(\mathcal{D}, \mathcal{B}_1) = 1]|$$

where $\Omega = (N, p_1, p_2, p_3, G, G_T, e)$, $s, \alpha \in \mathbb{Z}_N$, $g_1 \in G_{p_1}$, $\mathcal{X}_2, \mathcal{Y}_2, \mathcal{Z}_2 \in G_{p_2}$, $\mathcal{X}_3, \mathcal{Y}_3 \in G_{p_3}$, $\mathcal{B}_0 = e(g_1^\alpha, g_1^s)$ and $\mathcal{B}_1 \in_R G_T$.

Theorem 3. *If the algorithm \mathcal{G} satisfies Assumption 3, for any polynomial time adversary \mathcal{A} , its advantage $Adv_{\mathcal{A}}^3$ is negligible.*

Assumption 4. Let \mathcal{G} be the algorithm mentioned above and define a distribution tuple $\mathcal{D} = (\Omega, g_1 \mathcal{X}_2, g_1 \mathcal{X}_3, g_1^s \mathcal{Z}_3, \mathcal{Z}_2, g_3)$, then the advantage of \mathcal{A} in breaking the assumption is defined as:

$$Adv_{\mathcal{A}}^4 = |\Pr[\mathcal{A}(\mathcal{D}, \mathcal{B}_0) = 1] - \Pr[\mathcal{A}(\mathcal{D}, \mathcal{B}_1) = 1]|$$

where $\Omega = (N, p_1, p_2, p_3, G, G_T, e)$, $s \in \mathbb{Z}_N$, $g_1 \in G_{p_1}$, $\mathcal{X}_2, \mathcal{Y}_2, \mathcal{Z}_2 \in G_{p_2}$, $\mathcal{X}_3, \mathcal{Y}_3, \mathcal{Z}_3 \in G_{p_3}$, $\mathcal{B}_0 = g_1^s \mathcal{Y}_2 \mathcal{Y}_3$ and $\mathcal{B}_1 \in G_T$.

Theorem 4. *If the algorithm \mathcal{G} satisfies Assumption 4, for any polynomial time adversary \mathcal{A} , its advantage $Adv_{\mathcal{A}}^4$ is negligible.*

2.5 Algorithm Structure

An efficient policy hidden CP-ABE scheme is mainly composed of the following seven probabilistic polynomial time algorithms.

SetUp(1^λ) \rightarrow (Pp, Msk): The algorithm inputs λ and outputs a tuple (Pp, Msk), where λ , Pp and Msk are security parameter, system public parameters and master keys, respectively.

KeyGen(Pp, Msk, L) \rightarrow (Sk): This algorithm takes the public parameter Pp , the master key Sk , user attribute list $L = (N_L, V_L)$ as input and it outputs secret key Sk . It should be pointed out that N_L is the set of user attribute name, while V_L is the set of its value.

Encrypt($Pp, m, (M, \rho, T)$) \rightarrow (Ct): The algorithm inputs a tuple ($Pp, m, (M, \rho, T)$) and outputs ciphertext Ct . Where m is a plaintext message and

(M, ρ, \mathcal{T}) is the access policy defined by an encryptor and $\mathcal{T} = \{\tau_{\rho(1)}, \tau_{\rho(2)}, \dots, \tau_{\rho(m)}\}$ is a set of attribute value in this access policy.

Decrypt $(Pp, Sk, Ct) \rightarrow (m)$: This algorithm takes the public parameter Pp , the secret key Sk , the ciphertext Ct as input and it computes the decryption message m' and Ct' . If $Ct = Ct'$, it indicates that the message m' is valid and the algorithm outputs $m = m'$. Otherwise, it outputs the error symbol \perp .

3 Two Effective Attack Methods

In this section, we will introduce two effective methods to detect what access policy may be in a CP-ABE scheme.

3.1 Attack of Attribute Testing

Attribute testing is an efficient attack method, which mainly uses the correlation between ciphertext components and attribute associated with values and public parameters in a scheme, and then gives the corresponding attribute test group. Assume that there are n attributes in encryption system, and this test algorithm does at most n test calculations. The following is an example of an attribute testing and we only review the **Setup** and **Encrypt** algorithms of the Scheme [21].

Setup $(1^\lambda) \rightarrow (Pp, Msk)$: This algorithm inputs a security parameter 1^λ and outputs an initialized tuple $\Omega = (G, G_T, p, e, H, H_1)$, where p is a prime, $e : G \times G \rightarrow G_T$ is a bilinear map, G and G_T are cyclic group with p order, $H : G_T \rightarrow \{0, 1\}^*$ is a pseudo-random generator and $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_p$ is a collision resistant hash function. It also chooses $g, g_2, g_3, w, f, q, b \in G$, $\mu \in \mathbb{Z}_p$ and outputs the system public parameters $Pp = (\Omega, g, g_2, g_3, w, f, q, b, e(g, g)^\mu)$ and the system master keys $Msk = (\mu)$.

Encrypt $(Pp, m, \mathbb{A}) \rightarrow (Ct)$: The algorithm inputs the public parameter Pp , the data key $m \in \{0, 1\}^*$, an access policy \mathbb{A} and outputs the ciphertext Ct . It also chooses a random $\tau, \delta_1, \delta_2, \dots, \delta_l$ and sets a vector $\vec{v} = (s, z_2, z_3, \dots, z_n)$, where s is a secret value, $\{z_j\}_{j \in [2, n]}$ and $\{\delta'_j\}_{j' \in [1, l]}$ are random elements in \mathbb{Z}_p and $\tau \in \{0, 1\}^*$. Then the algorithm calculates the shared values $s_\phi = M_\phi \cdot \vec{v}$ for secret value s and the following ciphertext components. $E = (m || \tau) \oplus H(e(g, g)^{\mu s})$, $E_0 = g^s$, $E_{\phi, 1} = g^{s_\phi b^{\delta_\phi}}$, $E_{\phi, 2} = (w^{\varphi(\phi)} f)^{(-\delta_\phi)}$, $E_{\phi, 3} = g^{\delta_\phi}$, $cm = g_2^{H_1(m)} g_3^{H_1(\tau)}$.

Table 1. Some schemes are attacked by attribute testing

Schemes	Access control	Attribute testing
Zhang [9]	AND	$e(\widehat{C}_0, H(i v_{i,j})) = e(C_{i,j,\Delta}, g_1)$
Wang [20]	LSSS	$\rho(k) = v_{i,j}, \quad k \in [1, m]$
Fu [21]	LSSS	$e(E_{\phi,2}, g) = e(w^{v_{i,j}} f, E_{\phi,3}^{-1})$

Analysis of attribute testing: In the above scheme, $E_{\phi,2}$ is the only ciphertext component related to attribute, while $\{\delta_\phi\}_{\phi \in [1,l]}$ is a random value whose role is to serve as the blind factor of $E_{\phi,2}$. Generally speaking, it is impossible for an adversary to know the value of δ_ϕ randomly selected by the encryptor. An adversary does not know what attributes are embedded in the ciphertext, but it can construct a test pair $e(E'_{\phi,2}, E_{\phi,3}^{-1})$ by using the bilinear property of the map e , where $E'_{\phi,2} = w^\Delta f$ is a pseudo-ciphertext component constructed by the adversary and Δ is a possible attribute of adversary guess in access policy. Because w, f, g are the public parameter components, it is easy for the adversary to construct the test pair. Then the adversary uses the real ciphertext component $E_{\phi,2}$ and public key g to calculate the bilinear pair $\Theta = e(E_{\phi,2}, g)$. If the following equation is true,

$$e(E_{\phi,2}, g) = e(w^\Delta f, E_{\phi,3}^{-1})$$

it can determine that the ϕ^{th} attribute belongs to this access policy. Therefore, an adversary can determine all the attributes that make up this access policy by calculating l test pairs and n bilinear pairs at most, which is very deadly in smart medical cloud, smart home and other systems, since it seriously discloses the privacy of users. Moreover, some other schemes in Table 1 are also attacked by this method, in which the improvement scheme of [9] is given in [18]. We also emphasize that Wang's scheme [20] cannot really hide the access policy. Because ρ in their scheme directly maps the row label of M to an attribute and (M, ρ) was sent along with ciphertexts.

3.2 Guessing Attack of Access Policy

Policy guess attack is the another attack method, which is less efficient than the above attribute testing. Unlike attribute testing, this attack method emphasizes the whole idea, namely, an adversary first guesses a possible access structure and generates a test group with it, and then compares it with the other test group generated by normal ciphertexts and public parameters. If the comparison results of the two test groups are consistent, the adversary can successfully guess the access policy embedded in the ciphertexts. Assuming a policy hidden CP-ABE scheme was constructed by AND-gates on multi-valued attributes, which has n categories of attributes and each attribute has n_i possible candidate values, then it has a total of $\prod_{i \in [1,n]} n_i$ possible access policies. In order to describe this attack method more clearly, we review the scheme proposed by Waters [13]. The following are the **Setup** and **Encrypt** algorithms of their scheme.

Setup($1^\lambda, \mathcal{U}$) \rightarrow (PK, MSK): This algorithm takes a security parameter 1^λ as input and it outputs an initialized tuple $\Omega_1 = (G, G_T, p, e, g)$, where G and G_T are multiplicative cyclic groups with p order, \mathcal{U} is an attribute universe in this system, g is a generator of the group G and $e : G \times G \rightarrow G_T$ is a bilinear map. It also randomly selects $\theta, \eta \in \mathbb{Z}_p$ and $\{f_j\}_{j \in [1,n]} \in G$, where f_i is related to an attribute in \mathcal{U} . The system public key $PK = \langle \Omega_1, e(g, g)^\theta, g^\eta, \{f_j\}_{j \in [1,n]} \rangle$ and the master key $MSK = \langle g^\theta \rangle$.

Encrypt($PK, m, (A, \rho)$) \rightarrow (CT): This algorithm inputs the public key PK , a plaintext message m and an access policy (A, ρ) as input. It randomly selects $s, z_2, z_3, \dots, z_n \in \mathbb{Z}_p$ and sets a random column vector $\vec{v} = (s, z_2, \dots, z_n)^T$. It also calculates $\lambda_j = A_j \cdot \vec{v}$, where A_j denotes the j^{th} row of A and $\{\lambda_j\}_{j \in [1, l]}$ are the shared value of s . In addition, the algorithm selects random $\tau_1, \tau_2, \dots, \tau_l \in \mathbb{Z}_p$ and computes $C_0 = me(g, g)^{\theta s}$, $C_1 = g^s$, $C_j = g^{\lambda_j} f_{\rho(j)}^{-\tau_j}$, $D_j = g^{\tau_j}$. Therefore, the ciphertext for m is $CT = \langle C_0, C_1, \{C_j, D_j\}_{j \in [1, l]} \rangle$.

Table 2. Some schemes are attacked by access policy guessing

Schemes	Access control	Access policy guessing
Li [10]	AND	$e(C_{i,1}, \prod_{i \in \Delta} T_{i,j}) = e(X, \prod_{i \in W} C_{i,j,2})$
Waters [13]	LSSS	$\prod_{i \in \Delta} (e(T_i, g)e(W_i, u_{\rho(i)}))^{w_i} = e(C_2, g^a)$
Waters [23]	AND	$e(C_2, u' \prod_{i \in V} u_i) = e(C_3, g)$
Li [24]	LSSS	$\prod_{i \in \Delta} (e(C_x^{(1)}, g)e(C_x^{(2)}, T_x))^{w_i} = e(C^{(4)}, g^a)$

Analysis of access policy guessing: In this scheme, C_j is the only ciphertext component related to attribute, $\{\tau_j\}_{j \in [1, l]}$ are random value in \mathbb{Z}_p whose role is to blind the attribute correlation value f_i , while D_j is a ciphertext component directly associated with the blinding factor τ_j . The above three factors are obvious features of attribute or policy testing. Assuming that (A^*, ρ^*) is a guessing access policy, the adversary can calculate a index set \mathcal{I} through A^* and ρ^* . subsequently, the adversary also computes a constant set $\{\omega_j\}_{j \in [1, l]}$ by A^* and \mathcal{I} . Then the adversary can run the following policy testing.

$$\prod_{i \in \mathcal{I}} (e(C_j, g)e(D_j, f_{\rho(i)}))^{w_i} = e(C_1, g^\eta)$$

In order to address this problem, Wang *et al.* [3] proposed an improved scheme. Their scheme was constructed by composite order group and the main purpose of this idea is to use random elements in subgroup G_{p_3} to blind ciphertext component $\{C_1, C_j, D_j\}$. It is obvious from their work that this improvement can effectively resist the guessing attack of access policy. Unfortunately, their improved scheme failed to resist the attack of attribute testing because (A, ρ) had to be sent along with ciphertext in the encryption phase. An adversary can compute the following equation

$$\rho(k) = f_j, (k \in [1, l], j \in [1, n])$$

to determine which attributes belong to this access policy. Furthermore, some other schemes in Table 2 are also attacked by this method [23, 24], in which the improvement scheme of them were given in [19] and [25], respectively. Although the above method cannot determine the specific boolean expression of the access policy, the detected attributes are sufficient to reveal the users' privacy. Therefore, it is worth studying to design a hiding policy CP-ABE scheme that can resist the above two attacks at the same time.

3.3 Brief Summary

In a policy hidden CP-ABE algorithm, the policy defined by an encryptor is embedded in the ciphertext, namely, these ciphertext components $C_{i,j}$ are composed of attributes in the access policy. Generally speaking, some random elements r_i in \mathbb{Z}_p will be selected by encryptor to blind these ciphertext components, so as to hide the attributes in the policy. Moreover, in order to successfully decrypt, the encryptor has to add some ciphertext components $C_i = g^{r_i}$ similar to this one. Although r_i is randomly selected from \mathbb{Z}_p , the above attack method is still effective because the correlation between C_i and r_i and the characteristic of bilinear map. Therefore, it is necessary to eliminate this correlation and an effective way is to blind these ciphertext components $C_{i,j}$ twice. And its detailed description of the improved method is given in the proposed scheme in the Sect. 4.

4 An Efficient Policy Hidden CP-ABE Algorithm

In this part, we will describe the proposed scheme in detail. In our scheme, an attribute consists of attribute name and its value, which can solve the problem that Fu's [21] scheme faces the attack of attribute testing and most existing schemes face the guessing attack of access policy. We also pointed out that the use of random elements in subgroup G_{p_3} can resist the guessing attack of access policy and the use of $w^{\xi_{\rho(x)}}$ in the ciphertext component $C_{x,1}$ associated with the attribute can resist the attack of attribute testing. Furthermore, it also sets $\mathcal{E} = (\mathcal{E}_{enc}, \mathcal{E}_{dec})$ is a symmetric encryption scheme with a key space \mathcal{K} .

SetUp (1^λ) $\rightarrow (Pp, Msk)$: The algorithm takes a security parameter 1^λ as input and it outputs an initialized tuple $\Omega = (G, G_T, N, e, g, H)$, where $N = p_1 p_2 p_3$, $e : G \times G \rightarrow G_T$ is a bilinear map and H denotes a collision resistant hash function. It also randomly selects $g_0, h, w, f \in G$, $R \in G_{p_3}$, $\theta \in \mathbb{Z}_N$ and computes $\mathcal{Y} = e(g, g)^\theta$, $u = fR$. The system public parameters are $Pp = \langle \Omega, \mathcal{Y}, g_0, h, u, w \rangle$ and the master keys are $Msk = \langle \theta, f \rangle$.

KeyGen (Pp, Msk, L) $\rightarrow (Sk)$: The algorithm inputs in public parameters Pp , the master keys Msk , the user attribute list $L = (N_L, V_L)$ and output the user's secret keys Sk . Where $N_L \subset \{1, 2, \dots, n\}$ is the set of user attribute name and $V_L = \{\phi_1, \phi_2, \dots, \phi_{|N_L|}\}$ is the set of its attribute value. It also selects $\delta \in \mathbb{Z}_N$ for $i \in N_L$ and computes $K_0 = g^\theta w^\delta$, $K_1 = g^\delta$, $K_{i,1} = (g^{\phi_i} f)^\delta$. The user secret keys are $Sk = \langle K_0, K_1, \{K_{i,1}\}_{i \in N_L} \rangle$.

Encrypt ($Pp, m, (M, \rho, \mathcal{T})$) $\rightarrow (Ct)$: The encryption algorithm inputs a message m , the public parameters Pp , a access structure (M, ρ, \mathcal{T}) and outputs the ciphertexts Ct of m , where $\mathcal{T} = (\tau_{\rho(1)}, \tau_{\rho(2)}, \dots, \tau_{\rho(m)})$ ($\tau_{\rho(x)} \in \Psi_{\rho(x)}$) and M is a $m \times n$ access matrix. It first chooses $s, z_2, z_3, \dots, z_n \in \mathbb{Z}_N$ and sets a vector $\vec{v} = (s, z_2, z_3, \dots, z_n)$ and computes the share value $\xi_x = M_x \cdot \vec{v}$ belongs to the attribute $\phi_{\rho(x)}$, where M_x is the x^{th} row of M . Then it selects a symmetric encryption algorithm and calculates $E_0 = \mathcal{E}_{enc}(\mathcal{K}, m)$ and $E_1 = g_0^{H(m)} h^{H(\mathcal{K})}$. In addition, the algorithm also chooses $R_0, \{R_{x,1}, R_{x,2}\}_{x \in [1, m]} \in G_{p_3}$, $\{r_x\}_{x \in [1, m]} \in \mathbb{Z}_N$ and computes

$$C_0 = \mathcal{K}e(g, g)^{\theta s}, \quad C_1 = g^s \cdot R_0, \\ C_{x,1} = w^{\xi_{\rho(x)}}(g^{\tau_{\rho(x)}}u)^{-r_x} \cdot R_{x,1}, \quad C_{x,2} = g^{r_x} \cdot R_{x,2},$$

where $\mathcal{K} \in \mathcal{K}$ is the key of symmetric encryption algorithm \mathcal{E} . The ciphertexts Ct are published as: $Ct = \langle E_0, E_1, C_0, C_1, \{C_{x,1}, C_{x,2}\}_{x \in [1, m]} \rangle$.

Decrypt $(Ct, Sk, (M, \rho), L) \rightarrow (m)$: The algorithm takes the ciphertexts Ct , the user's secret keys Sk , the access matrix M , user's attribute list L as input and it outputs the message m . It first computes the index set Γ and the constant set $\mathbb{C} = \{\omega_i\}_{i \in \Gamma}$, where $\Gamma = \{i | \rho(i) \in N_L\}$ and \mathbb{C} can be calculated by M , ρ and Γ . Our decryption process is as follows.

$$\Delta = e(K_0, C_1) / \prod_{i \in \Gamma} (e(K_1, C_{i,1})e(K_{i,1}, C_{i,2}))^{\omega_i}, \\ \mathcal{K} = C_0 / \Delta, \\ m = \mathcal{E}_{dec}(\mathcal{K}, E_0).$$

The correctness of decryption is given as:

$$\Delta = \frac{e(K_0, C_1)}{\prod_{i \in \Gamma} (e(K_1, C_{i,1})e(K_{i,1}, C_{i,2}))^{\omega_i}} \\ = \frac{e((g^\theta w^\delta), g^s \cdot R_0)}{\prod_{i \in \Gamma} (e((g^\delta), w^{\xi_{\rho(i)}}(g^{\tau_{\rho(i)}}u)^{-r_i} \cdot R_{i,1})e((g^{\phi_{\rho(i)}}f)^\delta, g^{r_i} \cdot R_{i,2}))^{\omega_i}} \\ = \frac{e(g^\theta, g^s)e(w^\delta, g^s)}{\prod_{i \in \Gamma} (e(g^\delta, w^{\xi_{\rho(i)}})e(g^\delta, g^{-r_i \tau_{\rho(i)}})e(g^\delta, f^{-r_i})e(g^{\phi_{\rho(i)}}^\delta, g^{r_i})e(f^\delta, g^{r_i}))^{\omega_i}} \\ = e(g, g)^{\theta s}$$

Then the algorithm computes $\mathcal{K}' = C_0 / \Delta$, $m' = \mathcal{E}_{dec}(\mathcal{K}', E_0)$. If the following equation

$$Ct' = g_0^{H(m')} h^{H(\mathcal{K}')} = Ct$$

is true, it outputs the message $m = m'$, namely, decryption success. Otherwise, it outputs the error symbol \perp .

In our scheme, $C_{x,1}$ is the only ciphertext component related to attribute. $R_{x,1}$ and $R_{x,2}$ are the random elements in the subgroup G_{p_3} whose role is to blind the ciphertext components $C_{x,1}$ and $C_{x,2}$. In order to eliminate the correlation of blinding factor r_x between $C_{x,1}$ and $C_{x,2}$, we added the random component $w^{\xi_{\rho(x)}}$ to the ciphertext $C_{x,1}$. Therefore, an adversary fails to determine our defined access policy and its attributes by using the above attack methods.

5 Security Analysis

5.1 Security Model

In this section, the basic definition for choosing plaintext security is as follows and it is defined by security game between an adversary \mathcal{A} and a challenger \mathcal{B} .

Initialization: In this phase, two challenge access policy $\mathbb{M}_0 = (M, \rho, \mathcal{T}_0)$ and $\mathbb{M}_1 = (M, \rho, \mathcal{T}_1)$ are sent by the adversary to the challenger, then they play the following interactive games.

SetUp: The challenger first runs the algorithm and inputs a security parameter 1^λ , then the public parameters Pp is sent to the adversary and the master key Msk is secretly kept by the challenger.

Phase 1: In this phase, \mathcal{A} asks \mathcal{B} for the private keys of challenge attribute lists L_1, L_2, \dots, L_k . In response, \mathcal{B} first runs the algorithm **KenGen**, and then sends these private keys to the adversary. The restriction that none of these attribute lists can satisfy the challenge access policy.

Challenge: In this phase, two challenge plaintext message m_0 and m_1 was submitted by \mathcal{A} , where $|m_0| \neq |m_1|$. As a response, the challenger flips a random coin $b \in \{0, 1\}$ and the message m_b is encrypted by running the algorithm **Encrypt**, then the ciphertext Ct of m_b is sent to \mathcal{A} .

Phase 2: It is the same as phase 1. The adversary asks for the private keys for attribute lists $L_{k+1}, L_{k+2}, \dots, L_K$, however, none of them can satisfy the access policy defined in the initialization phase.

Guess: In this phase, the adversary outputs his guess bit $b' \in \{0, 1\}$. If $b' = b$, he will win the game, otherwise, challenge failure.

Definition 1: A policy hidden CP-ABE scheme is selective security if none of probabilistic polynomial time adversaries can break our security game with a negligible advantage ϵ , where $\epsilon = |Pr[b = b'] - \frac{1}{2}|$.

5.2 Our Proof

Our security proof employs the technique called dual system encryption, which is the same as Wang’s work. In a dual system, it usually has two semi-functional (SF) structures, namely, semi-functional ciphertexts and semi-functional keys. It is infeasible for the SF-keys to decrypt SF-ciphertexts, while SF-ciphertexts and normal ciphertexts can be decrypted by the normal keys. We also pointed out that the two SF-structures only used in the proof. Let g_2 be a generator of the subgroup \mathbb{G}_{p_2} .

SF-ciphertexts: It chooses random $\varsigma, v_1, \dots, v_m \in \mathbb{Z}_N$, computes $\tilde{C}_0 = C_0$, $\tilde{C}_1 = C_1 \cdot g_2^\varsigma$, $\tilde{C}_{x,1} = C_{x,1} \cdot g_2^{v_x}$, $\tilde{C}_{x,2} = C_{x,2}$ and sets the SF-ciphertexts as follows.

$$\tilde{Ct} = \langle \tilde{C}_0, \tilde{C}_1, \{\tilde{C}_{x,1}, \tilde{C}_{x,2}\}_{x \in [1,m]} \rangle.$$

SF-keys: It chooses random $\varpi_1, \varpi_2 \in \mathbb{Z}_N$, computes $\tilde{K}_0 = K_0 \cdot g_2^{\varpi_1}$, $\tilde{K}_1 = K_1 \cdot g_2^{\varpi_2}$, $\tilde{K}_{i,1} = K_{i,1}$ and sets the SF-keys as follows.

$$\tilde{Sk} = \langle \tilde{K}_0, \tilde{K}_1, \{\tilde{K}_{i,1}\}_{i \in \Gamma} \rangle.$$

Therefore, when a SF-keys is used to decrypt a SF-ciphertexts, the extra component $e(g_2, g_2)^{\varsigma \varpi_1 - \varpi_2 \sum_{i \in \Gamma} v_i \omega_i}$ will appear in the decryption phase. If $\varsigma \varpi_1 - \varpi_2 \sum_{i \in \Gamma} v_i \omega_i = 0$, it means that the SF-keys can successfully decrypt

the SF-ciphertexts and we also note that the SF-keys are special kind of normal keys.

Lemma 5. *The efficient policy hidden CP-ABE algorithm is chosen plaintext attack security under the Assumptions 1, 2, 3 and 4.*

The above theorem is proved by a series of games. Let Game_r be a real game, that is, its ciphertexts and keys are normal. Game_0 is the second game and its ciphertext is SF-structure. In $\text{Game}_{j'}$, the keys are modified to semi-functional form one by one, where $j' \in [1, k - 1]$. The challenge keys and the ciphertexts are modified to semi-functional form in Game_k . In Game_f , the message can be distinguished from a random message in the challenge ciphertexts. Finally, $C_{x,1}$ is a random element selected from the group G in Game_F . Therefore, after the end of a series of simulation games, the challenge ciphertexts is irrelevant to the access structure selected by the adversary.

Lemma 6. *Suppose there exists an algorithm \mathcal{A} such that $|\text{Game}_r \text{Adv}_{\mathcal{A}} - \text{Game}_0 \text{Adv}_{\mathcal{A}}| = \epsilon$, where ϵ is non-negligible value. Then the algorithm \mathcal{B} can be constructed with advantage ϵ in breaking Assumption 1.*

Proof. \mathcal{B} is given a tuple $(N, p_1, p_2, p_3, G, G_T, e, g_1, g_2)$ of assumption 1 and will simulate Game_r or Game_0 with \mathcal{A} . \mathcal{B} uniformly chooses $g_0, h, w, f \in G$, $R \in G_{p_3}$, $\theta \in \mathbb{Z}_N$ and sends the system public parameters

$$Pp = \langle \Omega, e(g, g)^\theta, g_0, h, fR, w \rangle$$

to \mathcal{A} . Consider $MsK = \langle \theta, f \rangle$, \mathcal{B} can answer the key extraction queries from \mathcal{A} in phase 1 because it know the master keys.

In challenge phase, \mathcal{A} sends two challenge messages m_0 and m_1 ($|m_0| = |m_1|$), and two challenge access structures (M, ρ, \mathcal{T}_0) and (M, ρ, \mathcal{T}_1) to \mathcal{B} . With restriction that the two challenge access structures can not be satisfied by any of the queried attribute sets in phase 1. \mathcal{B} also randomly chooses $\{b, c\} \in \{0, 1\}$, $d_1, \dots, d_l, r_1, \dots, r_l \in \mathbb{Z}_N$, $Q_0, \{Q_{x,1}, Q_{x,2}\}_{x \in [1, l]} \in G_{p_3}$ and outputs the challenge ciphertexts Ct^* associated with the challenge access policy (M, ρ, \mathcal{T}_c) as

$$\begin{aligned} C_0^* &= \mathcal{K}e(g^\theta, \mathcal{B}), & C_1^* &= \mathcal{B} \cdot Q_0, \\ C_{x,1}^* &= \mathcal{B}^{\tilde{\xi}_{\rho(x)}} (\mathcal{B}^{\tilde{\tau}_{\rho(x)}} (\mathcal{B}R))^{-r_x} \cdot Q_{x,1}, & C_{x,2}^* &= g^{r_x} \cdot Q_{x,2}. \end{aligned}$$

If $\mathcal{B} \leftarrow \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}$, let $\mathcal{B} = g^s g_2^\pi$, where π is a random element in \mathbb{Z}_N , then Ct^* are

$$\begin{aligned} C_0^* &= \mathcal{K}e(g^\theta, \mathcal{B}) = \mathcal{K}e(g^\theta, g^s) e(g^\theta, g_2^\pi) = \mathcal{K}e(g, g)^{\theta s} \\ C_1^* &= \mathcal{B} \cdot Q_0 = g^s Q_0 \cdot g_2^\pi, \\ C_{x,1}^* &= \mathcal{B}^{\tilde{\xi}_{\rho(x)}} (\mathcal{B}^{\tilde{\tau}_{\rho(x)}} (\mathcal{B}R))^{-r_x} \cdot Q_{x,1} \\ &= g^{s \tilde{\xi}_{\rho(x)}} (g^{s \tilde{\tau}_{\rho(x)}} (g^s R))^{-r_x} Q_{x,2} \cdot g_2^{\pi \tilde{\xi}_{\rho(x)} - (\tilde{\tau}_{\rho(x)} + 1) \pi r_x} \\ C_{x,2}^* &= g^{r_x} \cdot Q_{x,2}, \end{aligned}$$

where $f = g^s$, $w^{\xi_{\rho(x)}} = f^{\tilde{\xi}_{\rho(x)}}$ and $g^{\tau_{\rho(x)}} = f^{\tilde{\tau}_{\rho(x)}}$. It shows that the challenge ciphertexts is semi-functional and \mathcal{B} simulates Game_0 . If $B \leftarrow \mathbb{G}_{p_1}$, it is a normal ciphertext and \mathcal{B} simulates Game_r . Therefore, \mathcal{B} can break assumption 1 by using the output of \mathcal{A} , if \mathcal{A} can distinguish between Game_0 and Game_r with a non-negligible probability.

Lemma 7. *Suppose there exists an algorithm \mathcal{A} such that $|\text{Game}_{k-1}\text{Adv}_{\mathcal{A}} - \text{Game}_k\text{Adv}_{\mathcal{A}}| = \epsilon$, where ϵ is non-negligible value. Then the algorithm \mathcal{B} can be constructed with advantage ϵ in breaking Assumption 2.*

Lemma 8. *Suppose there exists an algorithm \mathcal{A} such that $|\text{Game}_k\text{Adv}_{\mathcal{A}} - \text{Game}_f\text{Adv}_{\mathcal{A}}| = \epsilon$, where ϵ is non-negligible value. Then the algorithm \mathcal{B} can be constructed with advantage ϵ in breaking Assumption 3.*

Lemma 9. *Suppose there exists an algorithm \mathcal{A} such that $|\text{Game}_f\text{Adv}_{\mathcal{A}} - \text{Game}_F\text{Adv}_{\mathcal{A}}| = \epsilon$, where ϵ is non-negligible value. Then the algorithm \mathcal{B} can be constructed with advantage ϵ in breaking Assumption 4.*

Since Lemmas 7, 8 and 9 are same as the proof of Lemma 6, we just give the detailed proof of Lemma 6. We also point out that the detailed process of our proof is the similar to [17] and [20].

Table 3. Performance comparison with other schemes

Schemes	Access control	Hidden policy	Security model
Lai [14]	LSSS	Yes	Subgroup assumption
Li [19]	AND-gates	Yes	DBDH
Our	LSSS	Yes	Subgroup assumption

Table 4. Comparisons of overhead with other schemes

Schemes	Public Parameters size	Ciphertexts size	Decryption cost
Lai [14]	$(n + 4) G + G_T $	$(4n + 2) G + 2 G_T $	$(2 I + 1)\mathbf{p} + I E_T$
Li [19]	$(n + 4) G + G_T $	$2 G + G_T $	$2\mathbf{p} + E_T$
Our	$5 G + G_T $	$(2n + 1) G + G_T $	$(2 I + 1)\mathbf{p} + I E_T$

6 Performance Comparison

In this part, we give some comparisons with other schemes, including the performance of our scheme, the size of public parameters and ciphertexts and the overhead of decryption as shown in Tables 3 and 4. Obviously, the overhead of public parameters and ciphertexts in our scheme are lower than that Lai's work with the same access structure. It notes that $|G|$ and $|G_T|$ are the number of bits of element in the group G and G_T respectively. \mathbf{p} denotes a paring operator and E_T denotes an exponentiation operation in G . n is the number of attributes in this system and $|I|$ is the size of the set satisfying the access structure.

7 Conclusion

In this paper, we analyze the existing policy hidden CP-ABE schemes and find out that most of them fail to effectively hide access policy. One of the main reasons is that the blinding factor of ciphertext components associated with attributes has strong correlation with other ciphertext components. Therefore, it is easy to detect the access policy embedded in the ciphertext or the attributes are related to access structure by using bilinear paring. In this paper, we introduce two effective attack methods based on these characteristics, namely, attack of attribute testing and guessing attack of access policy. In order to hide the access policy, we propose an effective improvement method. But our scheme was constructed in composite bilinear group. A prime order scheme will be given in our future work.

References

1. 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
2. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, Berkeley, CA, USA, pp. 1–14. IEEE (2007)
3. Goyal, V., Pandey, O., Sahai, A.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, pp. 89–98. ACM (2006)
4. Pirretti, M., Trayaor, P., Mcdaniel, P., Waters, B.: Secure attribute-based systems. In: Computer and Communications Security, New York, USA, pp. 99–112, ACM (2006)
5. Liu, L., Lai, J., Deng, R.H., Li, Y.: Ciphertext-policy attribute-based encryption with partially hidden access structure and its application to privacy-preserving electronic medical record system in cloud environment. *Secur. Commun. Netw.* **9**(18), 4897–4913 (2016)
6. Zhang, Y., Zheng, D., Deng, R.H.: Security and privacy in smart health: efficient policy-hiding attribute-based access control. *IEEE Internet Things* **5**(3), 2130–2145 (2018)
7. Nishide, T., Yoneyama, K., Ohta, K.: Attribute-based encryption with partially hidden encryptor-specified access structures. *Appl. Cryptogr. Netw. Secur.* **5037**(3), 13–23 (2009)
8. Malluhi Q.M., Shikfa A., Trinh V.C.: A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption. In: ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, pp. 230–240. ACM (2017)
9. Zhang Y., Chen X., Li J.: Anonymous attribute-based encryption supporting efficient decryption test. In: ACM SIGSAC Symposium on Information, Computer and Communications Security, Hangzhou, China, pp. 511–516. ACM (2013)
10. Li, J., Wang, H., Zhang, Y., Shen, J.: Ciphertext-policy attribute-based encryption with hidden access policy and testing. *KSII Trans. Internet Inf. Syst.* **10**(7), 3339–3352 (2016)

11. Emura, K., Miyaji, A., Nomura, A., Omote, K., Soshi, M.: A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In: Bao, F., Li, H., Wang, G. (eds.) ISPEC 2009. LNCS, vol. 5451, pp. 13–23. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00843-6_2
12. Cheung, L., Newport, C.: Provably secure ciphertext policy ABE. In: 14th ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA, pp. 456–465. ACM (2007)
13. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_4
14. Lai, J., Deng, R.H., Li, Y.: Expressive CP-ABE with partially hidden access structures. In: 7th ACM Symposium on Information, Computer and Communications Security, Seoul, Korea, pp. 18–19. ACM (2012)
15. Khan, F., Li, H., Zhang, L., Shen, J.: An expressive hidden access policy CP-ABE. In: IEEE Second International Conference on Data Science in Cyberspace, Shenzhen, China, pp. 178–186. IEEE (2017)
16. Yang, K., Han, Q., Li, H., Zheng, K., Shen, X.: An efficient and fine-grained big data access control scheme with privacy-preserving policy. *IEEE Internet Things J.* **4**(2), 563–571 (2016)
17. Zhang, L., Hu, G., Mu, Y.: Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system. *IEEE Access* **7**(1), 33202–33213 (2019)
18. Chaudhari, P., Das, M.L., Mathuria, A.: On anonymous attribute based encryption. *Inf. Syst. Secur.* **9478**(1), 378–392 (2015)
19. Li, X., Gu, D., Ren, Y., Ding, N., Yuan, K.: Efficient ciphertext-policy attribute based encryption with hidden policy. In: Xiang, Y., Pathan, M., Tao, X., Wang, H. (eds.) IDCS 2012. LNCS, vol. 7646, pp. 146–159. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34883-9_12
20. Wang, Z., He, M.: CP-ABE with hidden policy from waters efficient construction. *Int. J. Distrib. Sens. Netw.* **2016**(11), 1–8 (2016)
21. Fu, X., Nie, X., Wu, T., Li, F.: Large universe attribute based access control with efficient decryption in cloud storage system. *J. Syst. Softw.* **135**(4), 157–164 (2018)
22. Yin, H., Zhang, L., Cui, Y.: An improved ciphertext policy hiding attribute-based encryption with testing. *KSII Trans. Internet Inf. Syst.* **10**(7), 3339–3352 (2019)
23. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_7
24. Li, Q., Zhang, F.: A fully secure attribute based broadcast encryption scheme. *Int. J. Netw. Secur.* **17**(3), 263–271 (2015)
25. Cui, Y., Zhang, L.: Privacy preserving ciphertext-policy attribute-based broadcast encryption in smart city. *J. China Univ. Posts Telecommun.* **19**(1), 21–31 (2019)



Implementing Attacks on the Approximate Greatest Common Divisor Problem

Leizhang Wang^{1,2,3(✉)}, Quanbo Qu^{1,2}, Tuoyan Li³, and Yange Chen⁴

¹ State Key Laboratory of Integrated Service Networks, Xidian University,
Xi'an 710071, People's Republic of China

thisisleizhang@sina.cn

² Cryptographic Research Center, Xidian University,
Xi'an 710071, People's Republic of China

³ College of Applied Science, Beijing University of Technology,
Beijing 100124, People's Republic of China

⁴ School of Information Engineering, Xuchang University,
Xuchang 461000, China

Abstract. The security of many fully homomorphic encryption (FHE) schemes is guaranteed by the difficulty of the approximate greatest common divisor (AGCD) problem. Therefore, the study of AGCD problem is of great significance to the security of the fully homomorphic encryption. This paper surveys three kinds of attacks on the AGCD problem, i.e. exhaustive search attack, simultaneous Diophantine approximation (SDA) attack and the orthogonal lattice (OL) attack. We utilize the Number Theory Library (NTL) to implement the SDA attack and the optimized OL attack on the AGCD problem. Comparisons are performed based on the experimental results to illustrate that the exhaustive search attack can be easily defended just by increasing the size of ρ . And increasing the length of the public key is the most effective way to defend SDA attack and OL attack. Meanwhile, we concluded that the success rate of SDA attack and OL attack can be improved by increasing the dimension of lattice at the expense of a certain time efficiency. In addition, the analysis and experiments show that the fully homomorphic computing efficiency of FHE scheme can't be improved by simply increasing the private key without appropriately increasing the size of public key. Otherwise, the FHE scheme is vulnerable to OL and SDA attack. Besides, experimental results show that optimized OL attack performs better than both classical OL attack and SDA attack in terms of attack success rate and the time efficiency.

Keywords: Approximate greatest common divisor problem · Orthogonal lattice attack · Simultaneous diophantine approximation attack · Lattice reduction algorithm

1 Introduction

In 2009, Gentry [1] designed the first FHE scheme using ideal lattices, which is a breakthrough in this area. Subsequently, various FHE schemes were proposed, which were mainly divided into three categories according to different difficult assumptions:

FHEs based on ideal lattices [1, 4, 5, 10], FHEs based on the LWE (learning with errors) problem and its variants [3, 11, 21, 23], and FHEs based on the approximate greatest common divisor (AGCD) problem [2, 6–8, 20] firstly introduced by Howgrave-Graham [27] in 2001.

A simple approach to solving AGCD problem is exhaustive search on the error terms, but its computational complexity is exponential [22]. Simultaneous Diophantine approximation approach attack, one of the most efficient lattice attacks on AGCD problem was first proposed by Howgrave-Graham [27] in 2001. Then Dijk et al. further developed the lattice attack on AGCD problem by proposing orthogonal lattice attack [2] in 2010.

This paper mainly provides a comprehensive overview on the above three kinds of attacks on the AGCD problem, i.e., the exhaustive search attack, the simultaneous Diophantine approximation (SDA) attack, and the orthogonal lattice (OL) attack.

The paper is organized as follows: Sect. 2 gives some preliminaries on the AGCD problem, lattice and the SDA problem; Sect. 3 analyzes the time complexity of the exhaustive search attack on the AGCD problem; Sect. 4 mainly implements the SDA attack on the AGCD problem; Sect. 5 implements the optimized OL attack on the AGCD problem and Sect. 6 compares the SDA attack with the optimized OL attack based on our implementations.

2 Preliminaries

In this section, we firstly define the AGCD problem, and then introduce some concepts on lattices and lattice reduction algorithms, and finally give the definition and relevant conclusions of the SDA problem.

2.1 AGCD Problem

Given three positive integers γ, η, ρ with $\gamma > \eta > \rho$, the (γ, η, ρ) -AGCD problem is defined as follows:

For a random η -bit odd integer number p , given polynomially many instances,

$$\{a_i = pq_i + r_i : q_i \in \mathbb{Z} \cap (0, \frac{2^\gamma}{p}), r_i \in \mathbb{Z} \cap (-2^\rho, 2^\rho), 1 \leq i \leq n\},$$

output the approximate greatest common divisor p .

2.2 Lattices

An n -dimensional lattice L in an m -dimensional linear space can be spanned by n linearly independent m -dimensional row vectors b_1, \dots, b_n

$$L = \left\{ \sum_{i=1}^n k_i b_i \mid k_i \in \mathbb{Z}, 1 \leq i \leq n \right\}.$$

Where $\{b_1, \dots, b_n\}$ is a basis for lattice L , $B = [b_1^T, \dots, b_n^T]^T$ is the corresponding basis matrix. The rank or dimensions of the lattice L and determinant of L are respectively defined as $\dim L = n$ and $\det L = |\det(B)|$ if B is a square matrix.

2.3 Lattice Reduction Algorithm

Inputting a basis of a lattice, Lattice reduction algorithm can output a set of vectors as lattice reduction basis which is relatively short and almost orthogonal to each other. Lattice reduction algorithm has a wide application in cryptography. LLL algorithm [12] is the major lattice reduction algorithm used in this paper. Making a simple introduction to LLL algorithm: LLL lattice reduction algorithm is a polynomial time lattice reduction algorithm invented by Arjen Lenstra, Hendrik Lenstra and Laszlo Lovasz in 1982. Since there is no special polynomial time algorithm that can accurately solve the shortest vector problem in any high-dimensional space, LLL algorithm is used to approximate the shortest vector. Roughly speaking, LLL performs a continuous orthogonal projection, swapping two continuous vectors of the basis if needed, to get a reduced and nearly orthogonal basis.

Theorem 1. b_1, \dots, b_n is a LLL reduced basis of lattice L . When $\delta = \frac{3}{4}$, then

$$\|b_1\| \leq 2^{\frac{n-1}{2}} \|v\|, \forall v \in L, v \neq 0,$$

it is found that vector b_1 can be used to approximately replace the shortest non-zero vector in lattice L . In many cases b_1 is the shortest vector in the lattice. In addition, when the dimension of the lattice is lower, the shortest vector obtained by the LLL algorithm is closer to the shortest vector in the lattice [19].

2.4 Simultaneous Diophantine Approximation Approach

Simultaneous Diophantine approximation is a basic problem in Diophantine approximation theory, which has been widely used in fields such as cryptographic design [24] and analysis [25].

The problem is equivalent to finding a set of smaller fractions with the same denominator $p_1/q, p_2/q, \dots, p_n/q$ which approach $\alpha_1, \alpha_2, \dots, \alpha_n$ respectively, to make $|\alpha_i - p_i/q| \leq \varepsilon/Q$, where $\varepsilon, Q > 0$ and Q is an integer. When $Q \geq \varepsilon^{-n}$, the problem is solvable [26].

3 Exhaustive Search

The simplest way to solve the AGCD problem is exhaustive search of the noise terms directly. The p can be found by exhaustive search, if r_i is small enough that the $|r_i| < M$, where M is a fixed small integers, i.e. try every possible value of each r_i and r_2 , meanwhile check whether $\gcd(a_1 - r_1, a_2 - r_2)$ is a η -bit odd number. If not, then continue to update r_1 and r_2 . Such a search will eventually restore p .

Using the greatest common divisor fast algorithm, Stehlé-Zimmermann algorithm, solve the $\gcd(a_1 - r_1, a_2 - r_2)$. Considering r_i small enough (compared with a_i the examples of AGCD), $a_1 - r_1$ and $a_2 - r_2$ can be regarded as 2^γ bit number. So the time complexity of calculating is $O(2^\gamma)$. And because $r_i \in \mathbb{Z} \cap (-2^\rho, 2^\rho)$, using exhaustive search to solve AGCD problem time complexity is $O(2^{2\rho+\gamma})$. So when γ is more than 40, and ρ is greater than 20, exhaustive search will no longer be feasible.

In EUROCRYPT '12, Chen and Nguyen gave an algorithm that provides exponential acceleration on the basis of exhaustive search to solve AGCD problem [22]. This algorithm is essentially based on intelligent exhaustive search of noise terms through some polynomials. However, their approach requires a lot of memory. In their algorithms, they just need two elements from a series of elements x_i , and computational complexity of $O(2^{3\rho+\gamma})$. This means that if $\forall \gamma > 35, \rho > 30$ their algorithm will also be considered unworkable.

4 Simultaneous Diophantine Approximation Attack

4.1 Implementing SDA Attack on AGCD Problem

The basic idea of an SDA attack is to notice that: $a_i = pq_i + r_i, 0 \leq i \leq n$, where r_i is small relative to, then we have:

$$\frac{a_i}{a_0} \approx \frac{q_i}{q_0}$$

The fraction a_i/a_0 can be regarded as an approximation to unknown q_i/q_0 , By $\frac{a_i}{a_0} \approx \frac{q_i}{q_0}$, we get:

$$q_0 a_i - q_i a_0 \approx 0$$

Since $a_i = pq_i + r_i$, the above equation can be written as:

$$q_0 a_i - q_i a_0 = q_0 r_i - q_i r_0$$

In other words, the fraction a_i/a_0 is an instance of a simultaneous Diophantine approximation to unknown q_i/q_0 . If such a fraction q_i/q_0 can be determined, consider that r_0 is generally much smaller than q_0 , so we can recover p , by:

$$\left\lfloor \frac{a_0}{q_0} \right\rfloor = p + \left\lfloor \frac{r_0}{q_0} \right\rfloor$$

Constructing matrix M , based on $n + 1$ elements of AGCD problem:

$$M = \begin{pmatrix} 2^{\rho+1} & a_1 & \cdots & a_n \\ & -a_0 & & \\ & & \ddots & \\ & & & -a_0 \end{pmatrix}$$

Matrix M row vectors are the lattice bases of lattices L .

Let a vector in lattice L of which coordinates are (q_0, q_1, \dots, q_n) in the lattice base composed of row vectors of matrix M be \mathbf{v} , and \mathbf{v} is called the target vector. We have:

$$\begin{aligned} \mathbf{v} &= (q_0, q_1, \dots, q_n)M \\ &= (2^{\rho+1}q_0, q_0a_1 - q_1a_0, q_0a_2 - q_2a_0, \dots, q_0a_n - q_na_0) \\ &= (2^{\rho+1}q_0, q_0r_1 - q_1r_0, q_0r_2 - q_2r_0, \dots, q_0r_n - q_nr_0) \end{aligned}$$

From the previous discussion, we can see that each component of the target vector \mathbf{v} is very small, so naturally we hope that the length of the target vector \mathbf{v} is so small that \mathbf{v} is the shortest vector of lattice L . In this way, \mathbf{v} can be obtained by the lattice reduction algorithm. Then the question is whether \mathbf{v} is small enough to be the shortest vector of lattice L . The upper bound of \mathbf{v} is estimated below.

Find the upper bound of the length of the short vector \mathbf{v} in lattice L .

From $q_i \leftarrow Uni\{0, \dots, \lfloor p^{-1}2^\gamma \rfloor\}$ and $r_i \leftarrow Uni\{-2^\rho, \dots, 2^\rho\}$, where Uni represents uniform distribution and \leftarrow represents taking samples from the distribution. Calculating the second moments of q_i and r_i respectively, we have:

$$\begin{aligned} E(q_i^2) &= \int_{-\infty}^{+\infty} x^2 f(x) dx = \frac{p}{2^\gamma} \int_0^{p^{-1}2^\gamma} x^2 dx = \frac{2^{2\gamma}}{3p^2}, \\ E(r_i^2) &= \int_{-\infty}^{+\infty} x^2 f(x) dx = \frac{1}{2^{\rho+1}} \int_{-2^\rho}^{2^\rho} x^2 dx = \frac{2^{2\rho}}{3}, \\ E(r_i) &= 0. \end{aligned}$$

Considering that these random variables are independent, we can get:

$$\begin{aligned} E((q_0r_i - q_ir_0)^2) &= E(q_0^2r_i^2) + E(q_i^2r_0^2) - 2E(q_0r_iq_ir_0) \\ &= E(q_0^2)E(r_i^2) + E(q_i^2)E(r_0^2) - 2E(q_0q_i)E(r_ir_0) \\ &= \frac{2}{9} \cdot \frac{2^{2(\gamma+\rho)}}{p^2} \\ E(|\mathbf{v}|^2) &= \frac{2}{9} \cdot (n+1) \cdot \frac{2^{2(\gamma+\rho)}}{p^2} \end{aligned}$$

conclusion again by Jensen inequality: $[E(|\mathbf{v}|)]^2 \leq E(|\mathbf{v}|^2)$, thus get a more tight upper bound of the length of the shortest vector \mathbf{v} .

Since the basis matrix M of the special lattice L is upper triangular, the determinant of the special lattice L is easy to calculate, $\det(L) = 2^{\rho+1}a_0^n$. If short vector \mathbf{v} , as we hoped, is the shortest vector of the special lattice L , according to the Gaussian heuristic, we need to satisfy

$$\left[\frac{2(n+1)}{9}\right]^{\frac{1}{2}} \cdot 2^{\gamma+\rho-\eta} < (n+1)^{\frac{1}{2}} 2^{(\rho+1+\gamma n)/(n+1)}$$

Ignoring the constants in the above equation, the necessary but insufficient conditions for the success of SDA attack can be obtained:

$$n + 1 > \frac{\gamma - \rho}{\eta - \rho} \tag{1}$$

Therefore, when inequality (1) is established, the specific steps of SDA attack will be as follows: firstly, construct matrix M according to these instances of AGCD problem; secondly use lattice reduction algorithm, such as LLL algorithm, to find the approximation of shortest vector v in lattice L ; next calculate the coordinates of v under the base of M 's row vectors, which set as (q_0, q_1, \dots, q_n) and finally recover p by $\begin{bmatrix} a_i \\ q_i \end{bmatrix} = p + \begin{bmatrix} r_i \\ q_i \end{bmatrix}$.

4.2 Implementing SDA Attack on AGCD Problem by NTL Library Programming

$$n + 1 > \frac{\gamma - \rho}{\eta - \rho}. \tag{2}$$

It is clear from inequality (2) that increasing the number of instances n will make SDA attacks easier to success while increasing the length of public key γ will make SDA attacks more difficult to success. The results of mass of experiments are consistent with the theoretical speculation.

It can be clearly seen from Fig. 1 that under a certain value of γ , the increase of the number of instances is helpful to improve the success rate of SDA attack in the critical condition, which the attack is about to fail. However, promotion is not obvious when the public key is too small or too large. Additionally, when the number of examples is fixed, the increase of the value of γ will lead to the rapid decrease of the success rate of SDA attack.

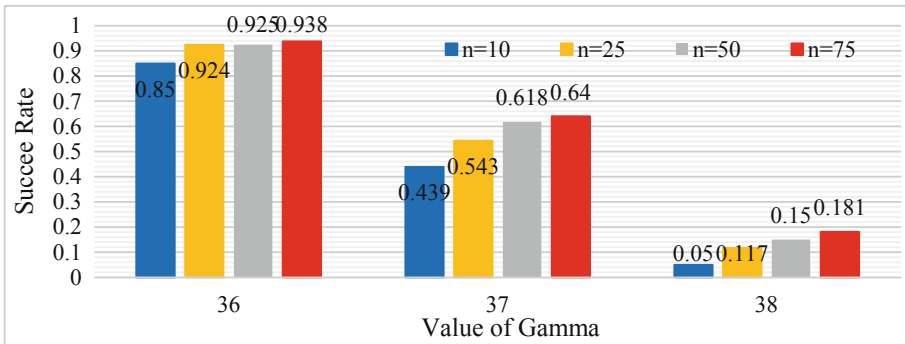


Fig. 1. The influence of value γ and n on the success rate of SDA attack

The reason is that when the value of γ increases too greatly, the length of target vector \mathbf{v} will become so large that target vector \mathbf{v} is no longer the shortest vector in lattice L . Therefore, the false assumption that \mathbf{v} is the shortest vector in lattice L leads to the failure of SDA attack.

It is obvious from inequality (2) that increasing the length of private key will make SDA attacks easier to success. The experimental results are consistent with the theoretical prediction.

Table 1. Effect of increased private key length on SDA attack success rate

Gamma	Eta = 12	Eta = 15	Rho	Gamma	Eta = 24	Eta = 26	Rho	Gamma	Eta = 28	Eta = 30	Rho
36	82.60%	100%	5	52	92.40%	100%	12	57	99.10%	100%	15
37	39.60%	100%	5	53	27.90%	100%	12	58	53.50%	100%	15
38	5%	100%	5	54	0%	100%	12	59	0%	100%	15
41	0%	99.20%	5	56	0%	91.50%	12	61	0%	99.00%	15
42	0%	82.90%	5	57	0%	30.40%	12	62	0%	55.00%	15

It is clear from Table 1 that SDA attacks will be more easily successful with the increase of the length (i.e. size) of the private key. This warns the designer of FHE scheme that the homomorphic computing efficiency of FHE scheme can't be improved by simply increasing the private key without appropriately increasing the size of public key. Otherwise, the security of the scheme is likely to be threatened by SDA attack. For example, when the length of the private key increases from 12 to 15, and the length of the public key remains unchanged at 39, the original scheme which almost can't be successfully attacked by SDA attack will be 100% likely to be broken.

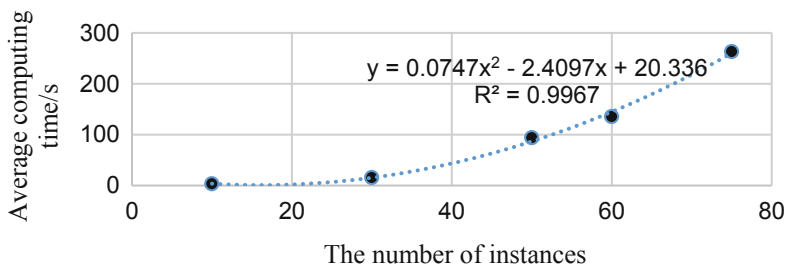


Fig. 2. Time consuming effects of increasing the number of instances

Figure 2 shows that linear increase in the number of instances will lead to a quadratic increase in the running time of SDA attack.

4.3 Summary

This section firstly transforms an AGCD problem with $n + 1$ instances into a simultaneous Diophantine approximation problem with n instances. Then, by constructing special lattices, the theoretical conditions for transforming the simultaneous Diophantine

approximation problem into finding the shortest vector problem on the lattice are obtained. Then the SDA attack on AGCD problem is implemented by using the lattice reduced algorithm in NTL library. Through theoretical analysis and experiments, it is proved that the most effective way to defend SDA attack is to increase the length of public key. SDA attack will not work if the public key is long enough. It warns the designer of FHE scheme that the homomorphic computing efficiency of FHE scheme can't be improved by simply enlarging the private key. Otherwise, simply increasing the length of the private key without properly increasing the length of the public key will make the FHE scheme vulnerable to SDA attack. At the expense of a certain time efficiency, by increasing the number of instances n in SDA attack, the attack success rate of SDA can be effectively improved in the critical case where attack is about to fail.

5 Orthogonal Lattice Attack

5.1 Review the Known Orthogonal Lattice Attacks

In this section, we outline the existing OL attack, where a_1, \dots, a_n are samples of (γ, η, ρ) -AGCD.

The OL attack for AGCD problem was firstly analyzed in [2] and then considered in [6, 27–30]. For a given n samples of AGCD, $a_i = pq_i + r_i$ there are three types of OL attack: the first is to consider the lattice orthogonal to (a_1, \dots, a_n) and (r_1, \dots, r_n) [2]; the second is to consider the lattice orthogonal to $(1, -r_1/2^\rho, \dots, -r_n/2^\rho)$ [2, 30, 31] and the third way is to consider the lattice orthogonal to $(1, -r_1, \dots, -r_n)$ [6, 28, 29]. In essence, the common point of these three OL attacks is to find vectors that are orthogonal to the unknown vector (q_1, \dots, q_n) .

5.2 Optimized Orthogonal Lattice Attack

For a (γ, η, ρ) -AGCD problem with n instances, our approach is as follows:

- First, we design a lattice to find these vectors which are orthogonal to the unknown vector (q_1, \dots, q_n) .
- Once we find enough of these vectors, we will recover (q_1, \dots, q_n) by solving the corresponding system of linear equations.
- Finally, considering $r_i < q_i$, we restore p by $p = \lfloor a_i/q_i \rfloor$.

5.3 Finding Vectors Orthogonal to (q_1, \dots, q_n)

For (γ, η, ρ) -AGCD problem with n samples, we define a lattice $L_2(\alpha)$, which determined by the parameter α . Lattice $L_2(\alpha)$ can be spanned by the row vectors of the following matrix $M(\alpha)$:

$$M(\alpha) = \begin{pmatrix} a_1 & \alpha & & & \\ a_2 & & \alpha & & \\ \vdots & & & \ddots & \\ a_n & & & & \alpha \end{pmatrix}$$

In particular, for lattice $L_2(\alpha)$, the row vectors of $M(\alpha)$ form a integer basis of $L_2(\alpha)$. And LLL lattice reduction algorithm can be used to obtain the lattice reduction basis v_1, \dots, v_n . Set $V = (v_1, \dots, v_n)^T$. We know that a unitary transformation is between two different lattice bases in the same lattice, therefore there is unimodular matrix U make:

$$V = UM(\alpha)$$

Next, we will find the theoretical condition that the row vectors of U matrix are the vectors which are orthogonal to unknown vector (q_1, \dots, q_n) .

We firstly give the following core lemma.

Lemma 1. Given the lattice reduction vector v in the lattice $L_2(\alpha)$, we have

$$\left| \sum_{i=1}^n u_i q_i \right| \leq \frac{\alpha + n^{\frac{1}{2}} 2^\rho}{\alpha} \cdot \frac{\|v\|}{2^{\eta-1}}.$$

Prove: According to $a_i = pq_i + r_i, 0 \leq i \leq n$, we have:

$$p \sum_{i=1}^n u_i q_i = \sum_{i=1}^n u_i a_i - \sum_{i=1}^n u_i r_i.$$

Setting $u = (u_1, \dots, u_n), r = (r_1, \dots, r_n)$, by triangle inequality and Schwarz inequality, we get

$$p \cdot \left| \sum_{i=1}^n u_i q_i \right| \leq \left| \sum_{i=1}^n u_i a_i \right| + \|r\| \cdot \|u\|.$$

Since $p \geq 2^{\eta-1}, |r_i| < 2^\rho, 1 \leq i \leq n$, namely $\|r\| \leq \sqrt{n} 2^\rho$. further, we have:

$$2^{\eta-1} \left| \sum_{i=1}^n u_i q_i \right| \leq \left| \sum_{i=1}^n u_i a_i \right| + n^{\frac{1}{2}} 2^\rho \cdot \|u\|. \quad (3)$$

Notice $v = \left(\sum_{i=1}^n u_i a_i, \alpha u_1, \dots, \alpha u_n \right)$ and $\alpha > 0$, so we have $\left| \sum_{i=1}^n u_i a_i \right| \leq \|v\|$ and $\|u\| \leq \frac{\|v\|}{\alpha}$. Substituting these two inequalities into Eq. (3), we complete the proof:

$$\left| \sum_{i=1}^n u_i a_i \right| \leq \frac{\alpha + n^{\frac{1}{2}} 2^\rho}{\alpha} \cdot \frac{\|v\|}{2^{\eta-1}}.$$

Notice that v is the basis vector for $L_2(\alpha)$. Setting v is the i -th basis of lattice $L_2(\alpha)$. Under the assumption of geometric series and the determinant of $L_2(\alpha)$ can be written as $L, \det L_2(\alpha) = \alpha^{n-1}(\alpha^2 + a_1^2 + \dots + a_n^2)$, so we have $L_2(\alpha) < (n+1)^{\frac{1}{2}} \alpha^{n-1} 2^\rho$, i.e.

$$\|v\| \leq \frac{(i+3)^{\frac{1}{2}}}{2} (n+1)^{\frac{1}{2n}} \delta_0^n 2^{\frac{\gamma-\rho}{n}-\eta}, \delta_0^n = \|b_1\| (\det L)^{-\frac{1}{n}}.$$

Then the boundary (3) is obtained directly from Lemma 5.2, namely:

$$\left| \sum_{i=1}^n u_i q_i \right| < (i+3)^{\frac{1}{2}} \frac{\alpha + n^{\frac{1}{2}} 2^\rho}{\alpha^{\frac{1}{n}}} (n+1)^{\frac{1}{2n}} \delta_0^n 2^{\frac{\gamma-\rho}{n}-\eta}. \tag{4}$$

Next, we minimize the upper bound of the $\left| \sum_{i=1}^n u_i q_i \right|$. Set $f(\alpha) = \frac{\alpha + n^{\frac{1}{2}} 2^\rho}{\alpha}$. Fix the value of γ, η, ρ and n .

As $f(\alpha)$ decreases, the upper bound becomes tighter. $f(\alpha)$ can be regarded as a Nike function. The derivative of $f(\alpha)$ is:

$$\begin{cases} f'(\alpha) < 0, & \text{when } 0 \leq \alpha \leq \alpha_0 \\ f'(\alpha) = 0, & \text{when } \alpha \leq \alpha_0 \\ f'(\alpha) > 0, & \text{when } \alpha > \alpha_0 \end{cases}.$$

Here $\alpha_0 = \frac{n^{\frac{1}{2}}}{n-1} 2^\rho$, we get $\min_{0 < \alpha} f(\alpha) = f(\alpha_0) = n \left(\frac{n^{\frac{1}{2}}}{n-1}\right)^{\frac{n-1}{n}} 2^{\frac{n-1}{n}\rho}$. Substitute $f(\alpha_0)$ into (4), then:

$$\left| \sum_{i=1}^n u_i q_i \right| < (i+3)^{\frac{1}{2}} g(n) \delta_0^n 2^{\frac{\gamma-\rho}{n}-\eta+\rho}.$$

Here $g(n) = n \left(\frac{n^{\frac{1}{2}}}{n-1}\right)^{\frac{n-1}{n}} (n+1)^{\frac{1}{2n}}$. And it can be proved that: $\lim_{n \rightarrow \infty} \frac{g(n)}{\sqrt{n}} = 1$.

Then, we get an asymptotic optimization bound, namely:

$$\left| \sum_{i=1}^n u_i q_i \right| < [n(i+3)]^{\frac{1}{2}} \cdot \delta_0^n 2^{\frac{\gamma-\rho}{n}-\eta+\rho}.$$

Finally, the conclusion is as follows.

Theorem 2. Set $v_i = \left(\sum_{j=1}^n u_{ij} a_i, \alpha u_{i1}, \dots, \alpha u_{in}\right)$ as the i -th basis of lattice $L_2(\alpha)$, $i = 1, \dots, n$. Take the optimum value of $\alpha = \frac{\sqrt{n}}{n-1} 2^\rho$. According to the geometric series hypothesis, we will get $\sum_{j=1}^n u_{ij} q_j = 0$, if (5) is true:

$$\frac{\gamma - \rho}{n} - (\eta - \rho) + n \log \delta_0 + \frac{\log n(i+3)}{2} < 0. \tag{5}$$

5.4 Restore q_1, \dots, q_n and P

We rewrite the system of linear equations:

$$\begin{cases} u_{1,1}q_1 + \dots + u_{1,n}q_n = 0 \\ \vdots \\ u_{n-1,1}q_1 + \dots + u_{n-1,n}q_n = 0 \\ u_{n,1}q_1 + \dots + u_{n,n}q_n = d \end{cases}$$

$$U \cdot (q_1, \dots, q_n)^T = (0, \dots, 0, d)^T. \tag{6}$$

The reason why the vector $(u_{n1}, u_{n2}, \dots, u_{nn})$ in the n -th line of linear equations is no longer orthogonal to vector (q_1, q_2, \dots, q_n) is that in n -dimensional vector space the orthogonal complementary space of vector (q_1, q_2, \dots, q_n) must be $n - 1$ dimensional. That is to say, $n - 1$ linear independent vectors orthogonal to vector (q_1, q_2, \dots, q_n) can be found at most. Besides, it can also be seen from inequality (5) that with the increase of i , inequality (5) will be difficult to hold. When i takes the maximum value n , the necessary condition that vector $(u_{n1}, u_{n2}, \dots, u_{nn})$ and vector (q_1, q_2, \dots, q_n) satisfy the orthogonality will no longer be valid, so the result of point multiplication of vector $(u_{n1}, u_{n2}, \dots, u_{nn})$ and vector (q_1, q_2, \dots, q_n) is not equal to 0, which is denoted as d .

From previous discussions, we know that the transition matrix U from the initial lattice basis matrix $M(x)$ to the lattice specification basis matrix V is a unimodular matrix. From the properties of unimodular matrix, we get $\det U = \pm 1$.

Moreover, the inverse of U : U^{-1} is also a unimodular matrix. For formula (6), pre-multiply it by U^{-1} to get:

$$(q_1, \dots, q_n)^T = U^{-1}(0, \dots, 0, d)^T.$$

Let $(w_{1n}, w_{2n}, \dots, w_{nn})^T$ be n -th column vector of U^{-1} , we can get:

$$(q_1, \dots, q_n) = d(w_{1n}, w_{2n}, \dots, w_{nn}).$$

This means that d is a common factor of q_1, \dots, q_n . However q_1, \dots, q_n are random integers drawing from $(0, 2^y/p)$. According to Euler product formula, we can prove the probability of $\gcd(q_1, q_2, \dots, q_n) = 1$ is $1/\zeta(n)$, here $\zeta(n)$ as Euler Riemann zeta function. And $\zeta(n)$ is a subtractive function of n , so with the increase of n , the probability of q_1, q_2, \dots, q_n mutual prime also increases. When $n = 4$, $1/\zeta(n)$ is about 92.39%, and in other words, when n isn't too small, the probability of overwhelming satisfies: $d = \pm 1$.

Therefore, we have:

$$(q_1, \dots, q_n) = (|w_{1n}|, |w_{2n}|, \dots, |w_{nn}|).$$

Here $w_{1n}, w_{2n}, \dots, w_{nn}$ are the elements of n -th column vector of U^{-1} .

Now we can recover p since we get q_1, \dots, q_n . Because $a_i = pq_i + r_i$, considering $|r_i| < |q_i|$ in most cases, we recover p by $\left\lfloor \frac{a_i}{q_i} \right\rfloor = p + \left\lfloor \frac{r_i}{q_i} \right\rfloor$.

5.5 Comparison of Optimized OL Attack and Classical OL Attack

Table 2. Comparison of optimized OL attacks and second OL attacks

n	ρ	η	γ	Experiments number	Success rate	Increased success rate	Time-consuming shortening
20	6	14	39	1000	99.5%	0.80%	5.23%
20	6	14	40	1000	72.4%	9.20%	7.95%
20	6	14	41	1000	21.2%	9.30%	6.70%
20	6	14	42	1000	0.01%	0.10%	10.32%
50	6	14	39	1000	100%	0.30%	28.34%
50	6	14	40	1000	71.5%	6.40%	29.09%
50	6	14	41	1000	22.8%	8.30%	30.51%
50	6	14	42	1000	0%	0%	45.35%

As can be seen from Table 2, compared with the classic second OL attack [2] proposed by Dijk et al., the success rate and time efficiency of the optimized OL attack are improved. The basic reason for the improvement of attack success rate is that the original constant 2^ρ is replaced by the optimum parameter α in the construction of the initial lattice base matrix, which makes the absolute value of $\left| \sum_{i=1}^n u_i q_i \right|$ smaller and the OL attack easier to succeed. The fundamental reason for the improvement of time efficiency is that the original lattice basis matrix is integrally excluded from the optimal parameter α , which reduces the size of each element in the original lattice basis matrix, thus reducing the computational complexity of the lattice reduction algorithm.

From Fig. 3 we can see that in the critical condition where Optimized OL attack is about to fail (the length of the public key is 40 and 41, respectively), the improvement of the success rate of the OL attack is the highest. When $ccn = 20, \rho = 6, \eta = 14, \gamma = 41$, the success rate of OL attack even increases by 9.3%. Secondly, when the dimension of lattice is small ($n = 20$), the success rate of attack is higher, because the smaller number of instances n in inequality (5) will make inequality (5) more tenable, which makes OL lattice attack easier to success.

As can be seen from Fig. 4, the improvement of time efficiency of optimized OL attack is not obvious in the case of low-dimensional attack, but it decreases significantly in the case of high-dimensional attack (the longer the public key is, the more obvious the reduction will be). When the dimension of lattice is 50 and the length of public key is 40, the success rate of optimizing OL attack is still 71.5%, and the time efficiency of optimizing OL attack is even improved by 29.09%.

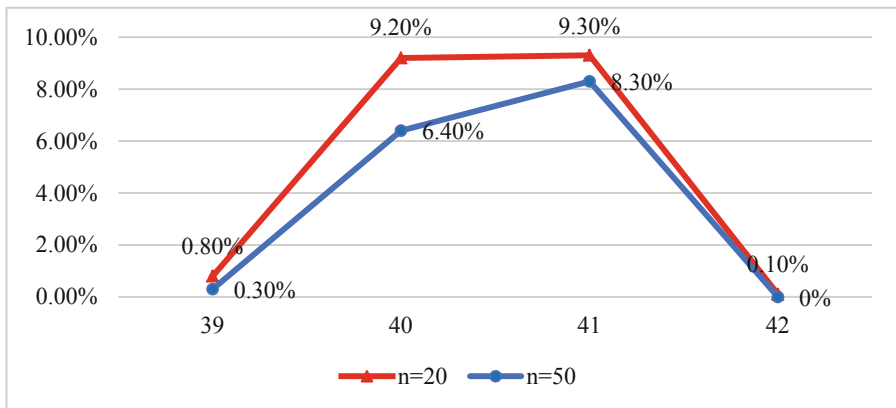


Fig. 3. Increase in the success rate of optimized OL attack under different n

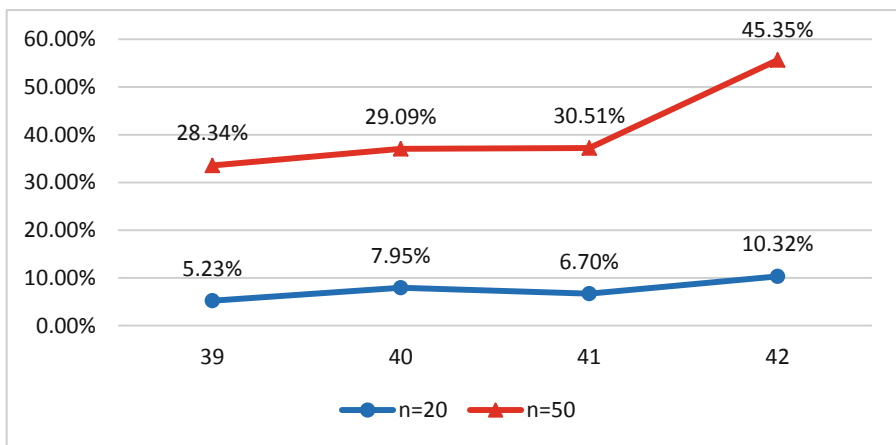


Fig. 4. Improvement of time efficiency in optimizing OL attack

5.6 Theoretical Analysis and Testing of OL Attacks

$$\frac{\gamma - \rho}{n} - (\eta - \rho) + n \log \delta_0 + \frac{\log n(i + 3)}{2} < 0. \tag{7}$$

As can be seen from inequality (7), the increase of public key length γ and number of instances n will make OL attacks more difficult to succeed. And the results of many times of experiments have proved that the theoretical results are consistent with the actual results. The specific results are shown in Table 3.

Table 3. The effect of the value of γ and n on the success rate of OL attack

Gamma	$n = 10$	$n = 50$	Rho	Eta	Experiments number
36	98.50%	100%	5	12	1000
37	69.90%	73.80%	5	12	1000
38	22.40%	21.70%	5	12	1000
52	99.60%	100%	12	24	1000
53	80.30%	94.60%	12	24	1000
58	98.50%	100%	15	28	1000
59	60.10%	93.10%	15	28	1000

Table 3 shows that, as with SDA attacks, increasing the number of instances will improve the success rate of OL attacks to some extent. Secondly, same with SDA attack, the value of public key length γ is still the key to defend OL attacks. When other security parameters are fixed and the values of γ are sufficiently large, OL attack will be unable to succeed.

Therefore, the larger the size of public key (controlled by γ) in FHE scheme based on AGCD problem is, the more resistant it will be to OL attack, but it will take more time to implement the scheme in practice. Literature [6] considers that when the size of public key reaches 2^{46} , it will be completely unable to apply to the practical system.

As can be seen from Table 4, the increase of private key length η will increase the success rate of the OL attack as well as the SDA attack when the other security parameters remain unchanged. It also reminds the designer of security scheme to increase the length of public key appropriately when improving the efficiency of FHE operation by increasing the length of private key. Otherwise, the scheme will be vulnerable to OL attack.

Table 4. Effect of η Value on OL attack success rate

Gamma	Eta = 12	Eta = 15	Rho	Experiments number
37	69.90%	100%	5	1000
38	22.40%	100%	5	1000
Gamma	Eta = 24	Eta = 25	Rho	Experiments number
53	76.80%	99.90%	12	1000
54	3.90%	99.50%	12	1000
Gamma	Eta = 26	Eta = 27	Rho	Experiments number
55	62.10%	99.70%	15	1000
56	1.90%	99.40%	15	1000

5.7 Summary

In this section, we give the theoretical conditions for the success of optimal OL attack. Using NTL library, the improved OL attack on AGCD problem is implemented. According to the test results, compared with the classical second OL attack, the success

rate and time efficiency of the optimized OL attack are improved. Meanwhile, it is reminded that the designer of security scheme should properly increase the length of public key when improving the efficiency of FHE operation by increasing the length of private key. Otherwise, the scheme will be vulnerable to OL attack. Besides, consistent with SDA attack, the increase of the dimension of the lattice will make the OL attack more easy to succeed.

6 Comparison Optimized OL Attack and SDA Attack on AGCD Problem

6.1 Comparison

In addition, according to Table 5, we can also know that under the same conditions of n, γ, η, ρ , when the length of public key is too big or too small, there is no significant difference in the success rate between SDA attack and OL attack, but in the critical case in which the attack is about to fail, the success rate of OL attack is higher than that of SDA attack.

Table 5. Effects of different attack modes on attack success rate

Gamma	OL	SDA	Rho	Eta	Experiments number
36	99.90%	94.20%	5	12	1000
37	73.60%	61.60%	5	12	1000
38	22.10%	13.30%	5	12	1000
54	99.50%	92.10%	12	25	1000
55	75.5%	25.30%	12	25	1000
56	99.40%	54.30%	15	27	1000
57	60.4%	0.50%	15	27	1000

Figure 5 shows more clearly that the success rate of the optimized OL attack is higher than that of the SDA attack in the extreme case in which the attack is about to fail. Specifically speaking, when the public key length is 36, 37 and 38, the success rate of the optimized OL attack is indeed higher than that of the SDA attack.

Figure 6 shows that the time efficiency of optimized OL attack is much higher than that of SDA attack.

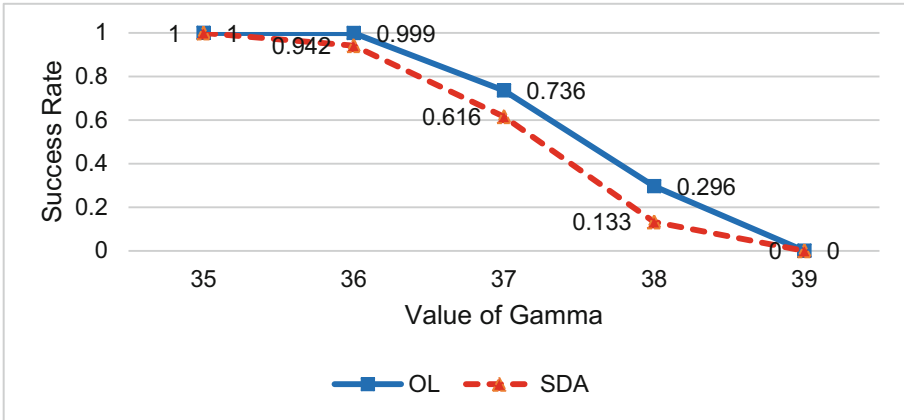


Fig. 5. Success rate of OL and SDA attacks under different conditions

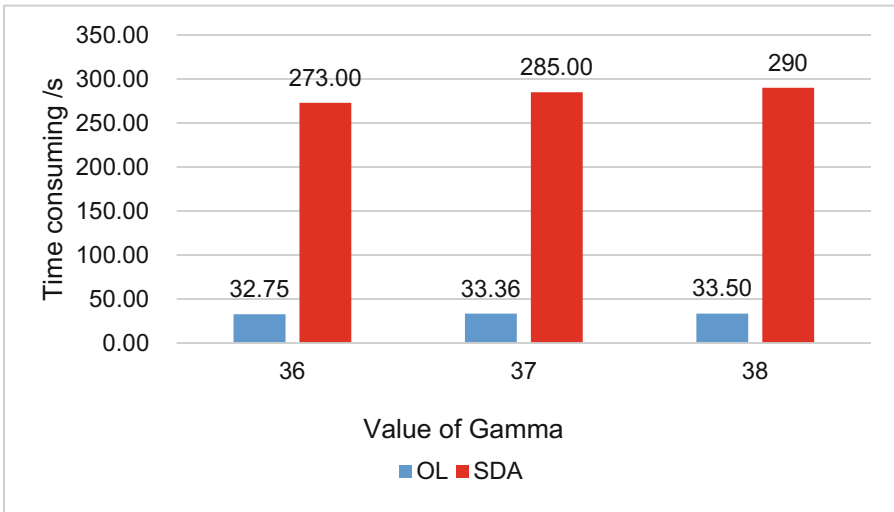


Fig. 6. Time-consuming of different attack modes

6.2 Summary

Optimized OL attack performs better than SDA attack in the terms of both attack success rate and time efficiency.

Acknowledgement. First of all, I would like to thank my mentor Professor Baocang Wang and Professor Hailou Yao. When I was puzzled to solve the AGCD problem, it was Professor Wang’s appropriate advice that guides me. In addition, when I wrote my paper, Professor Wang and Professor Yao also gave me many valuable opinions and suggestions which benefited me a lot. In the end, I would like to express my heartfelt thanks to Professor Wang and Professor Yao for their concern and help.

This work is supported by the National Key R&D Program of China under Grant No. 2017YFB0802000, the National Natural Science Foundation of China under Grant Nos. 61572390, U1736111, the National Cryptography Development Fund under Grant No. MMJJ20180111, the Plan For Scientific Innovation Talent of Henan Province under Grand no. 184100510012, the Program for Science & Technology Innovation Talents in Universities of Henan Province under Grant No. 8HASTIT022, the Innovation Scientists and Technicians Troop Construction Projects of Henan Province.

References

1. Gentry, C.: Fully homomorphic encryption using hidden ideal lattice. In: Proceedings of the 41st Annual ACM Symposium on Symposium on Theory of Computing-STOC 2009, pp. 169–178. ACM (2009)
2. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_2
3. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Foundations of Computer Science (FOCS). 2011 IEEE 52nd Annual Symposium on IEEE, 97–106 (2011)
4. Smart, N.P., Vercauteren, F.: Fully homomorphic encryption with relatively small key and ciphertext sizes. In: Nguyen, Phong Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 420–443. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_25
5. Stehlé, D., Steinfeld, R.: Faster fully homomorphic encryption. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 377–394. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_22
6. Coron, J.-S., Mandal, A., Naccache, D., Tibouchi, M.: Fully homomorphic encryption over the integers with shorter public keys. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 487–504. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_28
7. Coron, J.-S., Naccache, D., Tibouchi, M.: Public key compression and modulus switching for fully homomorphic encryption over the integers. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 446–464. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_27
8. Cheon, J.H., et al.: Batch fully homomorphic encryption over the integers. In: Johansson, T., Nguyen, Phong Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 315–335. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_20
9. Gentry, C., Halevi, S., Peikert, C., Smart, N.P.: Ring switching in BGV-style homomorphic encryption. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. LNCS, vol. 7485, pp. 19–37. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32928-9_2
10. Gentry, C., Halevi, S.: Implementing Gentry’s fully-homomorphic encryption scheme. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 129–148. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_9
11. Gentry, C., Halevi, S., Smart, Nigel P.: Fully homomorphic encryption with polylog overhead. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 465–482. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_28
12. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. *Math. Ann.* **261**(4), 515–534 (1982)

13. Schnorr, C.-P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.* **66**, 181–199 (1994)
14. Schnorr, C.P., Hörmel, H.H.: Attacking the chor-rivest cryptosystem by improved lattice reduction. In: Guillou, Louis C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 1–12. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-49264-X_1
15. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 31–51. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_3
16. Chen, Y., Nguyen, P.Q.: BKZ 2.0: better lattice security estimates. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 1–20. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_1
17. Novocin, A., Stehlé, D., Villard, G.: An LLL-reduction algorithm with quasi-linear time complexity: extended abstract. In: Proceedings of the Fortythird Annual ACM Symposium on Theory of Computing, STOC 2011, pp. 403–412. ACM, New York (2011)
18. Aono, Y., Wang, Y., Hayashi, T., Takagi, T.: Improved progressive BKZ algorithms and their precise cost estimation by sharp simulator. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 789–819. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_30
19. Meixia, L., Yunfei, F.: LLL algorithm and application. *J. Chongqing Vocat. Tech. Inst.* **16** (2), 161–163 (2007)
20. Chen, L., Ben, H., Huang, J.: An encryption depth optimization scheme for fully homomorphic encryption. In: International Conference on Identification, Information and Knowledge in the Internet of Thingsm Beijing, pp. 137–141 (2014)
21. Chen, Z., Wang, J., Zhang, Z., Song, X.: A fully homomorphic encryption scheme with better key size. *China Communications* **11**(9), 82–92 (2014)
22. Chen, Y., Nguyen, P.Q.: Faster algorithms for approximate common divisors: breaking fully-homomorphic-encryption challenges over the integers. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 502–519. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_30
23. Challa, R., VijayaKumari, G., Sunny, B.: Secure Image processing using LWE based Homomorphic encryption. In: IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). Coimbatore, pp. 1–6 (2015)
24. Baocang, W., Yupu, H.: Public key cryptosystem based on two cryptographic assumptions. *IEE Proc. Commun.* **152**(6), 861–865 (2005)
25. Baocang, W., Yupu, H.: Diophantine approximation attack on a fast public key cryptosystem. In: Chen, K., Deng, R., Lai, X., Zhou, J. (eds.) ISPEC 2006. LNCS, vol. 3903, pp. 25–32. Springer, Heidelberg (2006). https://doi.org/10.1007/11689522_3
26. Wang, B., Wu, Q., Hu, Y.: A knapsack-based probabilistic encryption scheme. *Inf. Sci.* **177** (19), 3884–3981 (2007)
27. Howgrave-Graham, N.: Approximate integer common divisors. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 51–66. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44670-2_6
28. Jintai, D., Chengdong, T.: A new algorithm for solving the general approximate common divisors problem and cryptanalysis of the FHE based on the GACD problem. *Cryptology ePrint Archive, Report 2014/042* (2014). <http://eprint.iacr.org/>
29. Lepoint, T.: Design and implementation of lattice-based cryptography. Theses, Ecole Normale Supérieure de Paris - ENS Paris, June 2014
30. Cheon, J.H., Stehlé, D.: Fully homomorphic encryption over the integers revisited. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 513–536. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_20

31. Galbraith, S.D., Gebregiyorgis, S.W., Murphy, S.D.: Algorithms for the approximate common divisor problem. In: Proceedings of Twelfth Algorithmic Number Theory Symposium (ANTS-XII) (2016)
32. Galbraith, S.D., Gebregiyorgis, S.W., Murphy, S.: Algorithms for the approximate common divisor problem. *LMS J. Comput. Math.* **19**(A), 58–72 (2016)
33. Xu, J., Sarkar, S., Hu, L.: Revisiting orthogonal lattice attacks on approximate common divisor problems and their applications. *Cryptology ePrint Archive: Report 2018/1208*, pp. 6–11 (2018)
34. Schnorr, C.P.: Lattice reduction by random sampling and birthday methods. *Stacs* **2607**, 145–156 (2005)



M4D: A Malware Detection Method Using Multimodal Features

Yusheng Dai¹(✉), Hui Li¹, Xing Rong², Yahong Li³, and Min Zheng⁴

¹ School of Electronics and Information,
Northwestern Polytechnical University, Xi'an, China
daiyusheng@mail.nwpu.edu.cn

² China Electric Engineering Design Institute, Beijing, China

³ School of Computer and Information Engineering,
Nan Yang Institute of Technology, Nanyang, China

⁴ Henan Institute of Information Security Co. Ltd, Xuchang, China

Abstract. With the increasing variants of malware, and it is of great significance to effectively detect malware and secure system. It is easy for malware to evade from the detection using existing dynamic detection method. To resolve the shortcomings of the existing dynamic detection method, we propose a multimodal malware detection method. By extracting the word vector of API call sequence conversion of malware, and extracting the image features converted from grayscale image memory dump of malware process, and inputting the multimodal features into the deep neural network is used to classify the malware samples. The effectiveness of this method is verified by the experiment through the captured malware samples in the wild. In addition, there is a performance comparison between our method and other recent experiments.

Keywords: Malware · Dynamic analysis · Memory dump · Multi-modal analysis

1 Introduction

The current number of malware has an explosive increase, seriously threatening personal information security and national security. Malware is secluded and antagonistic, which makes malware detection more difficult, so it is important to be able to effectively detect malware.

Malware detection is mainly divided into dynamic detection and static detection [11]. Static detection is vulnerable to confusing and cryptographic attack. In contrast, dynamic detection can monitor the execution of samples, thus avoiding the shortcomings of static detection and receiving extensive attention of researchers. However, dynamic detection is vulnerable to evasion attacks [6], resulting in a decline in detection performance.

At present, there are a large number of researches on malware dynamic detection based on software features, all of which can obtain good detection performance [4, 5, 9], but the detection fails when encountering malware with evasion

behavior. Smutz et al. [21] proposed a method of using mutual agreement analysis combined with ensemble learning for detection of evasion behavior, but the software itself contains the same defect density, making the detector equally vulnerable to attack [22], and using only software features is vulnerable to evade attack. Demme et al. [8] proposed to use hardware performance counters (HPC) as features for malware detection. On the basis of the above, Khasawneh et al. [14, 15] developed an integrated approach of using multiple HPC information for malware detection. However, due to the use of only hardware features, the behavior of malware can't be fully described and it is vulnerable to targeted evasion attacks.

In response to the above problems, we propose our **Multimodal Method for Malware Detection (M4D)** based on researches [7] and software features. The M4D method software features use the API call sequence and vectorize the call sequence; the hardware feature uses a memory dump file to map the memory dump file to a grayscale image and extract features from the texture of the grayscale image. A deep neural network is constructed, the two features are used as input to the neural network and the samples detected are classified.

This paper mainly contributes 3 points as follows:

- (1) We propose a new malware detection framework that uses API call sequences and the grayscale images mapped by the memory dump as input features for malware detection and describes malware from multiple dimensions. This method can ameliorate the problem of poor detection accuracy caused by malware evasion.
- (2) For both software and hardware features, we select a multimodal deep neural network. Choosing different feature extraction methods according to different characteristics can effectively enhance the malware classification accuracy.
- (3) We use the actual malware samples, so as to conduct experiments and evaluations on the methods described in this paper, and confirm the effectiveness of such methods.

The rest of the paper is as follows: Sect. 2 introduces related work; Sect. 3 details M4D method; Sect. 4 tests and evaluates M4D method; Sect. 5 summarizes the content of this paper.

2 Related Work

Common malware dynamic detection techniques using software features include behavior-based dynamic detection [9, 10, 23] and dynamic detection researches using API call sequence [4, 5], both of which can achieve relatively high detection rates. However, these researches focus on the detection accuracy and performance of the detector, and cannot effectively deal with the problem of dynamic detection evasion. Existing researches can effectively detect malware by category, but the emergence of malware variants (technical updates) will also lead to a decline in detection rate [12]. Another evasion attack method for dynamic detection is called mimicry attack [16, 17], which achieves the purpose of deceiving the

classifier through a series of filling useless codes or the addition of benign instructions. For evasion problem such as mimicry attack, Smutz et al. [21] proposed a method of using mutual agreement analysis combined with ensemble learning to detect mimicry attacks, but using only one feature will still be subject to other evasion attacks.

As the detection software may contain the same weakness as the normal application softwares, it is vulnerable to different degrees of evasion. Demme et al. [8] revealed that when the malware is running, the result of hardware performance counter information can be used to detect malware. Ozsoy and Khasawneh et al. [14, 15, 19, 20] proposed the use of a variety of low-level hardware features to improve the form of suspicious file weights, and enhance the detection of malware. However, malware behavior cannot be fully described due to using only low-level hardware features such as performance counters. Dai et al. [7] proposed that the use of memory dump grayscale images and the extraction of image textures into HOG features can effectively detect malware. Since memory storage information is more than hardware performance counters, the effective description of malware can be increased. Khasawneh et al. proposed RHMD [13], which enhances the detection of evasion attacks through integration of hardware features and the retraining of malicious samples not correctly classified. This kind of resilient detector can avoid the evasion attacks of malware to some extent, but using only hardware features has a problem of insufficient accuracy.

3 M4D Method

This paper proposes that the M4D method has three steps. The first step is to extract the malware features from the dynamic sandbox and preprocess the features. The second step is to input the various features into the classifier in the form of feature fusion; the last step is to output the detection result of the classifier to judge the maliciousness of the samples. The experimental process is shown in Fig. 1.

3.1 API Call Sequence Extraction

All the features in this paper were extracted by using the cuckoo sandbox [1], in which the API call sequence was extracted by the execution of sandbox monitoring samples and all executed API functions are recorded. We then used word2vec [18] to convert the extracted API sequence into a feature vector as one of the input to the classifier.

As different kinds of application has different API call sequences and different APIs were related, there were higher-frequency function call combinations in different kinds of malware. We extracted the API function names of all malicious and benign samples through the statistical sandbox, and took the 300 function names with the highest frequency as the vocabulary.

The word2vec model is a two-layer neural network that maps each word to a vector, and can represent the relationship between words and words.

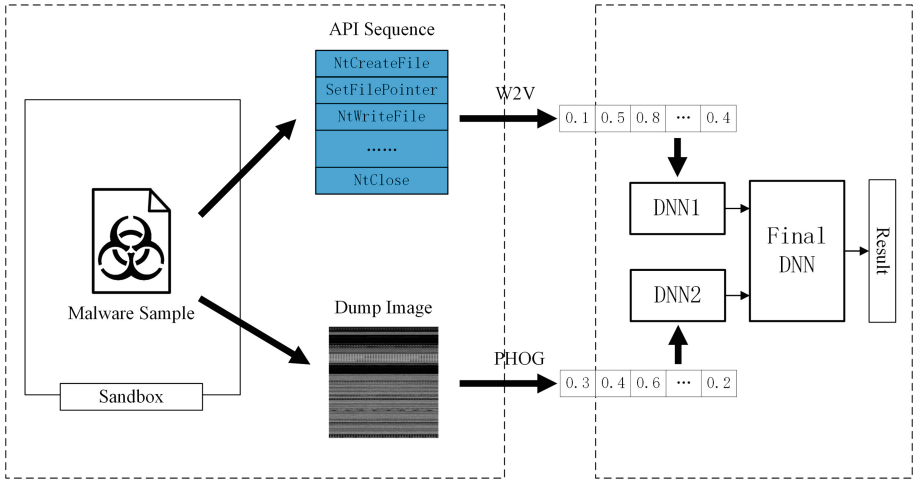


Fig. 1. M4D method framework

The word2vec method was to use the CBOW model and input the extracted API sequence into the word2vec model. The context word (function) was $c = 2$, and the word vector of the final sample is obtained by iterative calculation. We recorded the word vector $V_w = \{w_1, w_2, \dots, w_n\}$, where the function appearing in the sample produced a corresponding probability density, and the remaining length of the vector is padded with 0.

3.2 Memory Feature Extraction

The memory dump file of the sample process was used as a feature source, and improvements were made based on research [7]. Memory dumps were volatile data on physical memory and usually contained full memory dumps, core dumps, process dumps. We extracted complete process dump file from sandbox, and usually the full dump contents of a process was: dynamic link libraries (DLLs) associated with the process, environment variables, process heaps, thread stacks, data segments and text segments.

Usually the memory size occupied by process was different. We mapped every 8 bits of binary data in memory dump file to a $[0 - 255]$ grayscale pixel, and converted it to grayscale image according to fixed rowwidth and indefinite length. Since the column length of grayscale image after each dump conversion was inconsistent, in order to facilitate the preprocessing of subsequent features, bicubic interpolation was used to compress the grayscale image, so that all the converted images after processing became images with the same size. Figure 2 showed the memory dump grayscale image extracted by different families.

After the memory dump grayscale image with uniform size was obtained, the Pyramid Histogram of Gradient (PHOG) was used for image feature extraction. PHOG was a feature obtained by the calculation of Histogram of Gradient

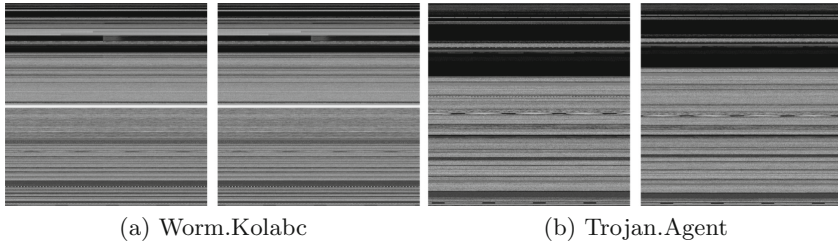


Fig. 2. Example of family grayscale image

(HOG) at different scales of the same picture. In this paper, we set a total of three levels, where the first level took X blocks, the second level took $4X$ blocks, and the third level took $16X$ blocks. Each block was normalized by L2-norm to vector feature.

3.3 Multimodal Neural Network

In M4D method, we used two features (API call sequence, memory dump PHOG) as the input of neural network, and the neural network was a multimodal deep neural network structure, i.e., the inputs of two features for initial network were independent of each other; After N layers of calculation, the output layers of two networks were merged into one vector as the input of final network. The network structure was shown in Fig. 3.

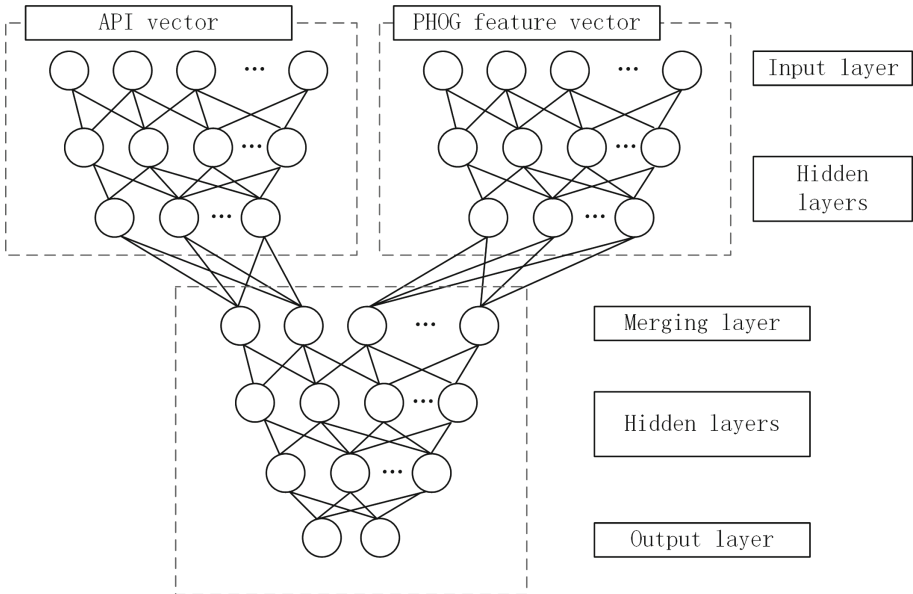


Fig. 3. Multimodal neural network structure

Our deep neural network model was similar to multilayer perceptron, with layers being fully connected, and using a rectified linear unit (ReLU) as activation function. In the model input layer, the vectors V_{api} and V_{mem} obtained through the extraction in Sects. 3.1 and 3.2 were used as inputs to the two initial networks. Each hidden layer consisted of several neurons. Each neuron was fully connected to the previous layer, and passed the data of this layer to the next layer through calculation. Each neuron output was expressed as:

$$x^l = \sigma(W^l x^{i-1} + b^l) \quad (1)$$

Where σ was the ReLU activation function, W^l was the weight matrix of the first layer, and b^l was the paranoid vector of the first layer. The current layer was calculated and the result was used as the input to the next layer. In the model, we used a dropout rate of 0.2 to prevent the generation of overfitting. The final output layer used softmax function regression, so as to convert the results calculated by neural network forward propagation into probability distribution.

4 Experimental Evaluation

4.1 Experimental Environment and Dataset

We used the computer environment with Intel (R) Core (TM) i5-6500 @3.20GHz CPU and 8GB DDR3 memory as the Host side of cuckoo sandbox to collect the dynamic behavior of sample. Where the Guest side of cuckoo sandbox was installed under the Ubuntu 16.04 system environment, using Windows 7 sp1 32-bit operating system and allocating 2 GB of memory.

M4D method ran on a graphics workstation. The hardware environment used Intel Core (TM) i7-6800K CPU, 32GB dual-channel DDR4 memory, and Geforce 1080Ti graphics card. The malware data used in this experiment came from OpenMalware [2], and the malware samples authorized for use were approximately 27k in total, collected between 2013 and 2015. For all samples, the family and type of malicious samples were detected and determined by VirusTotal [3]. The number of samples and information used in the experiment were shown in Table 1 in accordance with the type of malware.

4.2 Evaluation Criterion

According to the four values of true positive, false positive, true negative and false negative cases (TP, FP, TN, FN) generated in the experiment, the four indexes including accuracy rate (Acc), precision rate (P), recall rate (R) and F1-Score were calculated to measure classifier performance. The formulas of four indexes were expressed as follows:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

$$P = \frac{TP}{TP + FP} \quad (3)$$

Table 1. Dataset category

Sample category	Category No	Quantity
Backdoor	0	4652
Flooder	1	527
Worm	2	3821
Exploit	3	541
Trojan	4	2047
Adware	5	3226
Constructor	6	602
Hacktool	7	655
Other Malware	8	662
Benignware	9	1119

$$R = \frac{TP}{TP + FN} \quad (4)$$

$$F1 - Score = \frac{2 * P * R}{P + R} \quad (5)$$

Accuracy indicated the number of samples that were correctly classified in the classification process, precision indicated the percentage of positive examples correctly predicted to the positive examples actually predicted, and recall rate indicated the proportion of correct positive example in real positive examples. Precision and recall were mutually contradictory measurements. The use of F1-Score can balance precision rate and accuracy rate, while the closer the value was to 1, the better the performance.

4.3 M4D Method Evaluation

In the experiment, we used 80% of all available malicious samples and benign samples as the training set, and 20% of that as the test set. We used deep neural network algorithm, integrated algorithm (random forest) and multi-classified support vector machine to compare the same feature, so as to evaluate the contribution of each type of feature to detection performance in malware detection. As shown in Fig. 4, each feature used the precision rate, recall rate and F1-Score histogram of different classifiers.

As can be seen from Fig. 4, using API call sequence as the feature input classifier can achieve high detection precision, while DNN can obtain the highest detection precision and F1-Score among the three classifiers. The PHOG feature extracted by memory dump grayscale image was similar to API call sequence, and the use of DNN was also superior to the other two classifiers.

In general, the DNN method can obtain better classification performance no matter which feature was used. Perform fusion classification on the two features

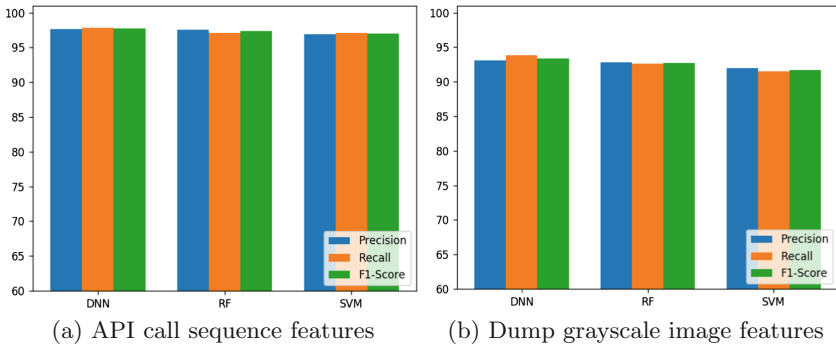


Fig. 4. Performance comparison of different classifiers

using our multimodal method, and compare in the aspect of detection accuracy, M4D method had higher accuracy than the method using single API call sequence and the method using single hardware feature. The comparison results were shown in Table 2.

Table 2. Experimental comparison of DNN features

Features	Accuracy	Precision	F1-Score
PHOG	93.3%	93.1%	93.5%
API	97.1%	97.3%	97.0%
M4D(two features)	97.2%	97.3%	97.2%

It can be seen from Table 2 that, in this set of experiments, the multimodal method can outperform the single feature in accuracy, and the hardware feature can be used as an auxiliary of software feature, so it can classify malware samples more accurately.

4.4 Evasion Detection

The same family of malware would produce several variants and continuously update, which was an evasion behavior itself [21]. Meanwhile, the detection classification algorithm would age [12], and the detection precision would also be reduced due to the newly generated malware. Based on these questions, we used the malware with generation time newer than training set and the malware with benign instructions inserted as the test samples.

In order to verify the effectiveness of our experimental method, we used the same dataset to compare the researches [15, 21], and took the optimal results of test, as shown in Table 3.

Table 3. Comparison of detected evasion behaviors

Method	Features	Accuracy	F1-Score
Smutz et al.	API	91.7%	92.4%
EnsambleHMD	HPC	88.2%	90.3%
M4D	API& PHOG	93.5%	93.9%

As can be seen from Table 3, the method of mixed feature was better than the other two methods. The research of Smutz et al. was based on the ensemble learning, using only API features, and the performance against new samples of the same family, semantic substitution and other escape modes was poor. The EnsambleHMD method proposed by Khasawneh et al. only used hardware performance counter as feature, which cannot fully describe malware behavior, resulting in insufficient detection precision.

5 Conclusion

The current dynamic analysis technology based on software features cannot effectively cope with the problem of malware evasion, and there is still some room for improvement for dynamic detection method based on hardware features in aspect of classification accuracy. This paper discusses the multimodal fusion methods using hardware and software features to detect malware. For these problems, we proposed to use the API call sequence and the memory dump grayscale as features, and use current deep neural network with well performance to multimodal fusion of the two features. By training different types of malware, it is verified that the method proposed in this paper can effectively detect malware and it has certain anti-evasion performance.

Acknowledgment. This work was supported by the National Natural Science Foundation of China under Grant 61571364, and Innovation Foundation for Doctoral Dissertation of Northwestern Polytechnical University under Grant CX201952.

References

1. Cuckoo sandbox. <https://cuckoosandbox.org/>
2. Openmalware. <http://malwarebenchmark.org/>. Last accessed 5 Apr. 2018
3. Virustotal. <https://www.virustotal.com/>
4. Sequence intent classification using hierarchical attention networks (March 2018). <https://www.microsoft.com/developerblog/2018/03/06/sequence-intent-classification/>

5. Athiwaratkun, B., Stokes, J.W.: Malware classification with LSTM and GRU language models and a character-level CNN. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2482–2486. IEEE (2017)
6. Bulazel, A., Yener, B.: A survey on automated dynamic malware analysis evasion and counter-evasion: Pc, mobile, and web. In: Proceedings of the 1st Reversing and Offensive-oriented Trends Symposium, p. 2. ACM (2017)
7. Dai, Y., Li, H., Qian, Y., Lu, X.: A malware classification method based on memory dump grayscale image. *Digital Invest.* **27**, 30–37 (2018)
8. Demme, J., et al.: On the feasibility of online malware detection with performance counters. In: ACM SIGARCH Computer Architecture News. vol. 41, pp. 559–570. ACM (2013)
9. Ding, Y., Xia, X., Chen, S., Li, Y.: A malware detection method based on family behavior graph. *Comput. Secur.* **73**, 73–86 (2018)
10. Hansen, S.S., Larsen, T.M.T., Stevanovic, M., Pedersen, J.M.: An approach for detection and family classification of malware based on behavioral analysis. In: 2016 International Conference on Computing, Networking and Communications (ICNC), pp. 1–5. IEEE (2016)
11. Idika, N., Mathur, A.P.: A survey of malware detection techniques. *Purdue University* **48** (2007)
12. Jordane, R., et al.: Transcend: detecting concept drift in malware classification models. In: 26th {USENIX} Security Symposium ({USENIX} Security 17), pp. 625–642 (2017)
13. Khasawneh, K.N., Abu-Ghazaleh, N., Ponomarev, D., Yu, L.: Rhmd: evasion-resilient hardware malware detectors. In: Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 315–327. ACM (2017)
14. Khasawneh, K.N., Ozsoy, M., Donovick, C., Abu-Ghazaleh, N., Ponomarev, D.: Ensemble learning for low-level hardware-supported malware detection. In: Bos, H., Monrose, F., Blanc, G. (eds.) RAID 2015. LNCS, vol. 9404, pp. 3–25. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26362-5_1
15. Khasawneh, K.N., Ozsoy, M., Donovick, C., Ghazaleh, N.A., Ponomarev, D.V.: Ensemblehmd: accurate hardware malware detectors with specialized ensemble classifiers. In: IEEE Transactions on Dependable and Secure Computing (2018)
16. Laskov, P., et al.: Practical evasion of a learning-based classifier: a case study. In: 2014 IEEE Symposium on Security and Privacy, pp. 197–211. IEEE (2014)
17. Maiorca, D., Corona, I., Giacinto, G.: Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious pdf files detection. In: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, pp. 119–130. ACM (2013)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013). arXiv preprint [arXiv:1301.3781](https://arxiv.org/abs/1301.3781)
19. Ozsoy, M., Donovick, C., Gorelik, I., Abu-Ghazaleh, N., Ponomarev, D.: Malware-aware processors: a framework for efficient online malware detection. In: 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA), pp. 651–661. IEEE (2015)
20. Ozsoy, M., Khasawneh, K.N., Donovick, C., Gorelik, I., Abu-Ghazaleh, N., Ponomarev, D.: Hardware-based malware detection using low-level architectural features. *IEEE Trans. Comput.* **65**(11), 3332–3344 (2016)
21. Smutz, C., Stavrou, A.: When a tree falls: using diversity in ensemble classifiers to identify evasion in malware detectors. In: NDSS (2016)

22. Tang, A., Sethumadhavan, S., Stolfo, S.J.: Unsupervised anomaly-based malware detection using hardware features. In: Stavrou, A., Bos, H., Portokalidis, G. (eds.) RAID 2014. LNCS, vol. 8688, pp. 109–129. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11379-1_6
23. Tobiyama, S., Yamaguchi, Y., Shimada, H., Ikuse, T., Yagi, T.: Malware detection with deep neural network using process behavior. In: 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC). vol. 2, pp. 577–582. IEEE (2016)



New Key Recovery Attack on the MICKEY Family of Stream Ciphers

Lin Ding^{1,2(✉)}, Dawu Gu¹, and Lei Wang^{1,3}

¹ Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
dinglin_cipher@163.com

² PLA SSF Information Engineering University, Zhengzhou 450001, China

³ Westone Cryptologic Research Center, Beijing 100000, China

Abstract. The well-known MICKEY 2.0 stream cipher, designed by Babbage and Dodd in 2006, is one of the seven finalists of the eSTREAM project. In this paper, new key recovery attack on the MICKEY family of stream ciphers in the single key setting is proposed. We prove that for a given variant of the MICKEY family of stream ciphers with a key size of $n(\geq 80)$ bits and a IV size of m bits, $0 < m < n$, there certainly exists a key recovery attack in the single key setting, whose online time, memory, data and offline time complexities are all smaller than 2^n . Take MICKEY 2.0 with a 64-bit IV as an example. The new attack recovers all 80 key bits with an online time complexity of 2^{78} , an offline time complexity of 2^{79} and a memory complexity of 2^{45} , requiring only 80 keystream bits. To the best of our knowledge, this paper presents the first cryptanalytic result of the MICKEY family of stream ciphers better than exhaustive key search.

Keywords: Cryptanalysis · Key recovery attack · MICKEY · Stream cipher

1 Introduction

The ECRYPT Stream Cipher Project, also known as eSTREAM, is a multi-year effort running from 2004 to 2008 to promote the design of efficient and compact stream ciphers suitable for widespread adoption. After a three-phase elimination process, the final eSTREAM portfolio was announced in September 2008 as a result of the project. The eSTREAM portfolio ciphers fall into two profiles. Profile 1 currently contains four stream ciphers, i.e., HC-128 [1], Rabbit [2], Salsa20/12 [3] and SOSEMANUK [4], which are suitable for software applications with high throughput requirements. Profile 2 currently contains three

This work was supported by the National Natural Science Foundation of China under Grant 61602514, 61802437, 61272488, 61202491, 61572516, 61272041, 61772547, National Cryptography Development Fund under Grant MMJJ20170125 and National Postdoctoral Program for Innovative Talents under Grant BX201700153.

stream ciphers, i.e., Grain v1 [5], MICKEY 2.0 [6] and Trivium [7], which are particularly suitable for hardware applications with restricted resources such as limited storage, gate count, or power consumption.

MICKEY [8] is a synchronous bit-oriented stream cipher designed by Babbage and Dodd for low hardware complexity and high speed. After a Time-Memory-Data tradeoff (TMDTO) attack [9] on the initial version of MICKEY (denoted as MICKEY 1.0), the designers modified it to be the current version, denoted as MICKEY 2.0. The revisions in MICKEY 2.0 had been precisely targeted at addressing the issues raised in [9]. In fact, the MICKEY family of stream ciphers currently consist of two variants, i.e., MICKEY 2.0 with an 80-bit key and MICKEY-128 2.0 [6] with a 128-bit key. MICKEY supports a variable IV in length, i.e., between 0 and 80 bits for MICKEY 2.0, and between 0 and 128 bits for MICKEY-128 2.0. The name of MICKEY stands for Mutual Irregular Clocking KEYstream generator which describes the behavior of the stream ciphers accurately. The internal state consists of two feedback shift registers named R and S , each of which is irregularly clocked and controlled by the other.

It's twelve years since the MICKEY family of stream ciphers were proposed. However, only some preliminary attacks [10–14] on them were available in literature, due to its strong cryptanalytic resistance. Up to now, there are no attacks better than exhaustive key search on the MICKEY family of stream ciphers. Besides these attacks, some side channel attacks on the MICKEY family of stream ciphers were published, e.g., differential fault analysis [15–18], correlation power analysis [19], differential power analysis [20] and scan-based side channel attack [21]. As the designers had claimed in [6], the MICKEY family of stream ciphers are standing up very well against classical cryptanalysis and provides a high level of security.

Unlike the most modern stream ciphers which initialize the initial state using all key and IV bits, the MICKEY family of stream ciphers initializes the initial state using all zeros, and in the first $m + n$ initialization steps utilize only one IV or key bit per initialization step to update the state. Here, m and n (≥ 80) denote the IV size and key size, respectively. Based on this important observation, a new key recovery attack on the MICKEY family of stream ciphers in the single key setting is proposed. We prove that the new attack is certainly better than exhaustive key search for $0 < m < n$, i.e., the time, memory and data complexities are all smaller than 2^n . Take MICKEY 2.0 with a 64-bit IV as an example. The new attack recovers all 80 key bits with an online time complexity of 2^{78} , an offline time complexity of 2^{79} and a memory complexity of 2^{45} , requiring only 80 keystream bits. To the best of our knowledge, the new attack is much better than all previous attacks on the MICKEY family of stream ciphers, and is the first attack better than exhaustive key search on the MICKEY family of stream ciphers.

The organization of the paper is as follows. A brief description of the MICKEY family of stream ciphers is given in Sect. 2. Previous attacks on the MICKEY family of stream ciphers are reviewed in Sect. 3. In Sect. 4, we will present our new key recovery attack on the MICKEY family of stream ciphers. Section 5 concludes the paper.

2 Brief Description of the MICKEY Family of Stream Ciphers

A detailed description of the MICKEY family of stream ciphers is available in [6]. The exact structure of the MICKEY family of stream ciphers is explained in Fig. 1. The keystream generator is built from two registers R and S , each of which is irregularly clocked and controlled by the other. Each register consists of l bits. We label the bits in the registers r_0, \dots, r_l and s_0, \dots, s_l , respectively.

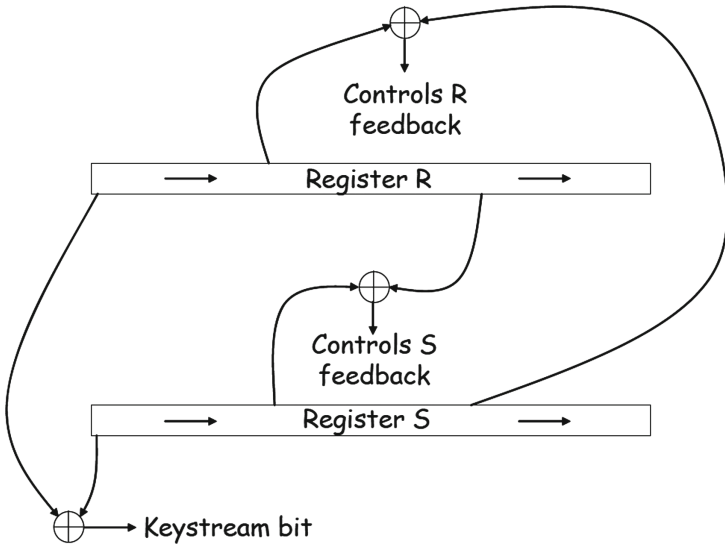


Fig. 1. The structure of the MICKEY family of stream ciphers

The state registers R and S are updated by an operation $\text{CLOCK_KG}(R, S, \text{MIXING}, \text{INPUT_BIT})$, which is defined as follows:

$$\begin{aligned}
 &\text{CLOCK_KG}(R, S, \text{MIXING}, \text{INPUT_BIT}) \\
 &\{ \\
 &\quad - \text{If } \text{MIXING} = \text{TRUE}, \\
 &\quad \cdot \text{CLOCK_R}(R, \text{INPUT_BIT_R} = \text{INPUT_BIT} \oplus s_a, \text{CONTROL_BIT_R} \\
 &= s_b \oplus r_c) \\
 &\quad - \text{If instead } \text{MIXING} = \text{FALSE}, \\
 &\quad \cdot \text{CLOCK_R}(R, \text{INPUT_BIT_R} = \text{INPUT_BIT}, \text{CONTROL_BIT_R} = \\
 &s_b \oplus r_c) \\
 &\quad - \text{CLOCK_S}(S, \text{INPUT_BIT_S} = \text{INPUT_BIT}, \text{CONTROL_BIT_S} = s_c \oplus \\
 &r_{b-1}) \\
 &\}
 \end{aligned}$$

Since our attacks are not concerned with the details of `CLOCK_R` and `CLOCK_S`, thus here we omit the detailed descriptions, which can be found in [6].

Before the generation of the first keystream bit, the initialization process should be executed, which is shown as follows.

- Initialize the registers R and S with all zeros.
- (IV Loading) For $0 \leq i \leq m - 1$:
 - `CLOCK_KG` ($R, S, \text{MIXING} = \text{TRUE}, \text{INPUT_BIT} = iv_i$)
- (Key Loading) For $0 \leq i \leq n - 1$:
 - `CLOCK_KG` ($R, S, \text{MIXING} = \text{TRUE}, \text{INPUT_BIT} = k_i$)
- (Prelock) For $0 \leq i \leq p - 1$:
 - `CLOCK_KG` ($R, S, \text{MIXING} = \text{TRUE}, \text{INPUT_BIT} = 0$)

After the initialization process, the cipher generates keystream bits z_0, \dots, z_{L-1} as follows:

- For $0 \leq i \leq L - 1$:
 - $z_i = s_0 \oplus r_0$
 - `CLOCK_KG` ($R, S, \text{MIXING} = \text{FALSE}, \text{INPUT_BIT} = 0$)

The selections of the parameters used in MICKEY 2.0 and MICKEY-128 2.0 are listed in Table 1.

Table 1. The selections of the parameters used in MICKEY 2.0 and MICKEY-128 2.0

	l	a	b	c	m	n	p
MICKEY 2.0	100	50	34	67	$0 \leq m \leq 80$	80	100
MICKEY-128 2.0	160	80	54	106	$0 \leq m \leq 128$	128	160

3 Previous Attacks on the MICKEY Family of Stream Ciphers

The MICKEY family of stream ciphers has a graceful structure which led many cryptanalysts to try to attack it during the past twelve years. Previous attacks on the MICKEY family of stream ciphers are summarized as follows.

Hong and Kim [9] showed that BSW sampling could be performed on MICKEY 1.0. They presented a Time-Memory-Data tradeoff attack on it with an online time complexity of 2^{66} , an offline time complexity of 2^{100} , a memory complexity of 2^{67} , and a data complexity of 2^{60} . In the attack, the online time, data and memory complexities are all less than 2^{80} . However, the offline time complexity is greater than 2^{80} . As a response, the designers tweaked the design by increasing the state size from 160 to 200 bits and altering the values of some control bit tap locations. When applying TMDTO attacks to MICKEY 2.0, at least one of T , M and D must be no less than 2^{80} , since the tradeoff

curve $TM^2D^2 = N^2$ has to be satisfied. Thus, the designers claimed that the MICKEY family of stream ciphers are immune to TMDTO attacks.

In [10], Tischhauser presented a new approach to the cryptanalysis of symmetric algorithms based on non-smooth optimization. When applying it to MICKEY 2.0, this method can solve instances corresponding to the full cipher. However, the time complexity is greater than exhaustive key search.

In [11], a slide resynchronization attack on MICKEY 2.0 and MICKEY-128 2.0 was proposed under the assumption that some related IVs of different lengths with the same key are allowed for the cryptanalysis. However, as claimed by the designers, it is not acceptable to use two IVs of different lengths with the same key. Thus, the assumption is not reasonable, which makes the attack impractical.

In [12], Helleseth et al. analyzed how to recover the secret key of MICKEY 2.0, assuming that the attacker, in some way, knows the internal state of the keystream generator at some step during initialization or keystream generation and knows exactly how many steps the generator was stepped to end up in this state. Clearly, the assumption is so harsh, which makes the attack also impractical.

In [13], Khoo and Tan claimed that they proposed a new TMDTO attack that can break eSTREAM ciphers and block cipher standards with both offline and online time complexities faster than exhaustive key search. Their idea is to break up the available online data complexity into two parts: D_{IV} to be the number of IV resynchronizations, and D_{single} to be the number of keystream bits available for each IV. They applied their attack to MICKEY 2.0. Unfortunately, their attack is incorrect, since multiple data points from the same keystream output can not be utilized when inverting the function $f : (Key \rightarrow Output Prefix)$, as pointed out in [22]. Therefore, their attack on MICKEY 2.0 is also incorrect.

In [14], Ding et al. proposed a new TMDTO attack on stream ciphers, by combining the time-memory-data tradeoff attack with the BSW sampling technique. They applied the attack to MICKEY 2.0. The results show that the online time complexity is smaller than 2^{80} , while the offline time, data and memory complexities are all no less than 2^{80} . A similar result on MICKEY-128 2.0 was also presented.

The results above show that there are no attacks with all complexities smaller than 2^n on the MICKEY family of stream ciphers to this day, due to its strong cryptanalytic resistance. As the designers had claimed in [6], the MICKEY family of stream ciphers is standing up very well against classical cryptanalysis and provides a high level of security.

4 New Key Recovery Attack on MICKEY Family of Stream Ciphers

In this section, we will present a new key recovery attack on the MICKEY family of stream ciphers. Recall the initialization process of the MICKEY family of stream ciphers. Unlike the most modern stream ciphers which initialize the initial state using all key and IV bits, the MICKEY family of stream ciphers

initializes the initial state using all zeros, and before Preclock one key or IV bit per initialization step is utilized to update the internal state. This style of designing the initialization process makes it possible for the attacker to mount a new key recovery attack on the MICKEY family of stream ciphers.

For convenience, we define a new process, called **M-Preclock**, which is obtained by adding one step to the p -step Preclock. That is, M-Preclock consists of $p + 1$ steps, as shown below:

- (M-Preclock) For $0 \leq i \leq p$:
 - CLOCK_KG ($R, S, \text{MIXING} = \text{TRUE}, \text{INPUT_BIT} = 0$)

Now, we construct a special one-way function as follows.

$$f' : ((n - 1) - \text{bit key segment} \rightarrow (n - 1) - \text{bit keystream segment})$$

When $1 < m < n$, the special function $f' : \{0, 1\}^{n-1} \rightarrow \{0, 1\}^{n-1}$ is obtained as follows.

-
1. Set $iv_i = 0$ ($0 \leq i \leq m - 2$) and $k_0 = 0$.
 2. Given an $(n - 1)$ -bit input value x , treat x as the $n - 1$ key bits k_1, \dots, k_{n-1} , execute IV Loading, Key Loading using x , and then M-Preclock, finally generating an $(n - 1)$ -bit keystream segment y .
 3. Output y .
-

Now, we will describe our new key recovery attack on the MICKEY family of stream ciphers. The attack relies on an assumption that the attacker knows the length of IV in advance which will be utilized to perform encryptions in the online phase. It is known that the above assumption is reasonable, since IV is freely chosen for the attacker in the chosen IV setting. The new attack consists of two phases, i.e., the offline phase and the online phase. A graphical explanation of our new key recovery attack can be found in Fig. 2.

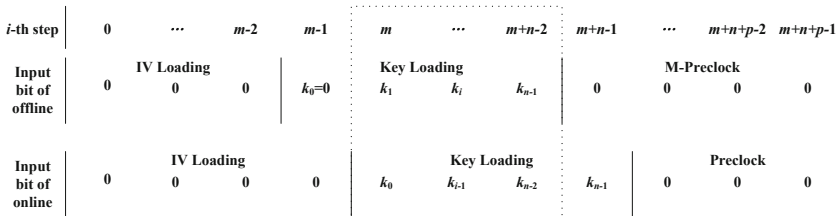


Fig. 2. A graphical explanation of the new key recovery attack

The details of the offline and online phases are described as follows.

In the offline phase, the attacker sets $iv_i = 0$ for all $0 \leq i \leq m - 2$ and $k_0 = 0$. In order to invert the special one-way function f' , the attacker executes

a pre-computation of original time-memory trade-off attack (by Hellman [23]) to construct Hellman Tables which covers the whole search space $N' = 2^{n-1}$. Since the initialization process of the MICKEY family of stream ciphers consists of many steps (i.e., no less than 180 steps for MICKEY 2.0 and no less than 288 steps for MICKEY-128 2.0), adding one step does not make significant change in the time complexity. Thus, we ignore the cost of replacing the Preclock with the M-Preclock when constructing the Hellman Tables.

In the online phase, the attacker waits for the special m -bit $IV = \mathbf{0}$ (which means $iv_i = 0$ for all $0 \leq i \leq m - 1$) to occur. After $IV = \mathbf{0}$ occurs, the attacker guesses $k_{n-1} = 0$, and applies the time-memory trade-off attack to invert the special function f' . Note that the input of f' here is made up of the $n - 1$ key bits k_0, \dots, k_{n-2} , instead of k_1, \dots, k_{n-1} in the offline phase, see the dotted rectangle in Fig. 2. If the time-memory trade-off attack succeeds to recover the value of the $n - 1$ key bits k_0, \dots, k_{n-2} , it implies that the guess is correct, i.e., $k_{n-1} = 0$ holds. Otherwise, if the time-memory trade-off attack fails, then it implies that the guess is incorrect, i.e., $k_{n-1} = 1$ holds, and then the $n - 1$ key bits k_0, \dots, k_{n-2} should be recovered by an exhaustive search.

The complexities of the proposed key recovery attack are calculated as follows.

In the offline phase, the attacker has to execute a pre-computation of original time-memory trade-off attack to construct Hellman Tables. Thus, the total offline time complexity is $P = N' = 2^{n-1}$. The memory used is denoted as M' .

In the online phase, the attacker has to wait the special m -bit $IV = \mathbf{0}$ to occur, which leads to a time complexity of 2^m for waiting. After the special IV is observed, the attacker proceeds to execute a time-memory trade-off attack with a time complexity of T' . Recall the tradeoff curve of the time-memory trade-off attack, it has

$$T' M'^2 = N'^2 = 2^{2(n-1)}$$

The online time complexity varies from the value of the last key bit k_{n-1} . If $k_{n-1} = 0$ holds, the attacker is able to recover the remaining $n - 1$ key bits k_0, \dots, k_{n-2} by the time-memory trade-off attack. Thus, the time complexity for $k_{n-1} = 0$ (denoted as T_1) is completely determined by the time complexity of the time-memory trade-off attack, i.e., $T_1 = T'$. If $k_{n-1} = 1$ holds, the attacker can not recover the remaining $n - 1$ key bits k_0, \dots, k_{n-2} by the time-memory trade-off attack, and has to recover them by an exhaustive search. Thus, the time complexity for $k_{n-1} = 1$ (denoted as T_2) is calculated as $T_2 = T' + 2^{n-1}$. Considering the time complexity for waiting for $IV = \mathbf{0}$ to occur, if each key bit is treated as a random independent variable, the expected online time complexity (denoted as T) is calculated as

$$T = 2^m + \frac{1}{2} \cdot T_1 + \frac{1}{2} \cdot T_2 = 2^m + \frac{1}{2} \cdot (T' + T' + 2^{n-1}) = 2^m + T' + 2^{n-2}$$

The total memory complexity (denoted as M) is completely determined by the memory used in the offline phase, i.e., $M = M'$. It is easy to see that the data complexity (denoted as D) of the new attack is quite small, i.e., $D = n$ bits, since

the attacker only requires the known keystream to verify whether the recovered n key bits are correct or not. Obviously, the new attack has one remarkable advantage in the data complexity when comparing it with other cryptanalysis techniques.

Up to now, we have mounted a new key recovery attack on the MICKEY family of stream ciphers, with an offline time complexity of $P = 2^{n-1}$, an expected online time complexity of $T = 2^m + T' + 2^{n-2}$, a memory complexity of $M = M'$, and a data complexity of $D = n$ bits. The tradeoff curve of $T'M'^2 = 2^{2(n-1)}$ should be satisfied for the new attack. Note that the new attack also works when $m = 1$ holds. The only difference from the attack described above is that when $m = 1$ holds, the Step 1 for constructing the special function f' only sets $k_0 = 0$, since no IV bits are used to construct the special function at this moment. Thus, the new attack works for $0 < m < n$. Usually, n is no less than 80 for a modern stream cipher to provide a high security level, i.e., $n \geq 80$ holds. A theorem is obtained as follows.

Theorem 1. For a given variant of the MICKEY family of stream ciphers with a key size of $n(\geq 80)$ bits and a IV size of m bits, $0 < m < n$, there certainly exists a key recovery attack in the single key setting, whose online time, memory, data and offline time complexities are all smaller than 2^n .

Proof. Recall the trade-off curve of $T'M'^2 = 2^{2(n-1)}$ in our new attack. Hence, a reasonable choice of T' and M' is $T' = M' = 2^{2(n-1)/3}$, then we have

$$M = M' = 2^{2(n-1)/3} < 2^n$$

Since $n \geq 80$, it implies $2^{2(n-1)/3} < 2^{n-2}$, then we know

$$T = 2^m + 2^{2(n-1)/3} + 2^{n-2} \leq 2^{n-1} + 2^{2(n-1)/3} + 2^{n-2} < 2^n$$

Clearly, both the offline time complexity of $P = 2^{n-1}$ and the data complexity of $D = n$ bits are smaller than 2^n .

Thus, this Theorem follows directly. ■

Now, we will apply our new key recovery attack to MICKEY 2.0 and MICKEY-128 2.0, and then the complexities are calculated as follows. For MICKEY 2.0, it has $n = 80$, and our attack is applicable for $0 < m < 80$.

For example, when $m = 64$ holds, choose $M = M' = 2^{45}$, then we have

$$P = 2^{79} \text{ and } T = 2^{64} + 2^{68} + 2^{78} \approx 2^{78}$$

For example, when $m = 79$ holds, choose $M = M' = 2^{45}$, then we have

$$P = 2^{79} \text{ and } T = 2^{79} + 2^{68} + 2^{78} \approx 2^{79.58}$$

For MICKEY-128 2.0, it has $n = 128$, and our attack is applicable for $0 < m < 128$.

For example, when $m = 96$ holds, choose $M = M' = 2^{69}$, then we have

$$P = 2^{127} \text{ and } T = 2^{96} + 2^{116} + 2^{126} \approx 2^{126}$$

For example, when $m = 127$ holds, choose $M = M' = 2^{69}$, then we have

$$P = 2^{127} \text{ and } T = 2^{127} + 2^{116} + 2^{126} \approx 2^{127.58}$$

As shown in Table 2, it compares the best known attacks on the MICKEY family of stream ciphers with our new attacks. The results show that our new key recovery attacks are much better than the best known attacks on the MICKEY family of stream ciphers, and are the first attacks certainly better than exhaustive key search in terms of all complexity indexes.

Table 2. Comparisons of our new attacks with the best known attacks on the MICKEY family of stream ciphers

Ciphers	Attacks	T	M	D	P
MICKEY 2.0	[14]	2^{47}	2^{120}	280 bits	2^{120}
MICKEY 2.0	This paper ($m = 64$)	2^{78}	2^{45}	80 bits	2^{79}
MICKEY 2.0	This paper ($m = 79$)	$2^{79.58}$	2^{45}	80 bits	2^{79}
MICKEY-128 2.0	[14]	2^{74}	2^{192}	2128 bits	2^{192}
MICKEY-128 2.0	This paper ($m = 96$)	2^{126}	2^{69}	128 bits	2^{127}
MICKEY-128 2.0	This paper ($m = 127$)	$2^{127.58}$	2^{69}	128 bits	2^{127}

5 Conclusions

It’s twelve years since the MICKEY family of stream ciphers was proposed. However, there are no attacks better than exhaustive key search on the MICKEY family of stream ciphers up to now, due to its strong cryptanalytic resistance. This paper proposes a new key recovery attack on the MICKEY family of stream ciphers in the single key setting. We prove that for a given variant of the MICKEY family of stream ciphers with a key size of $n(\geq 80)$ bits and a IV size of m bits, $0 < m < n$, there certainly exists a key recovery attack in the single key setting, whose online time, memory, data and offline time complexities all smaller than 2^n . To the best of our knowledge, this paper presents the first cryptanalytic result of the MICKEY family of stream ciphers better than exhaustive key search.

Acknowledgment. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

1. Wu, H.: The stream cipher HC-128. In: Robshaw, M., Billet, O. (eds.) New Stream Cipher Designs. LNCS, vol. 4986, pp. 39–47. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68351-3_4

2. Boesgaard, M., Vesterager, M., Zenner, E.: The rabbit stream cipher. In: Robshaw, M., Billet, O. (eds.) *New Stream Cipher Designs*. LNCS, vol. 4986, pp. 69–83. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68351-3_7
3. Bernstein, D.J.: The Salsa20 family of stream ciphers. In: Robshaw, M., Billet, O. (eds.) *New Stream Cipher Designs*. LNCS, vol. 4986, pp. 84–97. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68351-3_8
4. Berbain, C., et al.: Sosemanuk, a fast software-oriented stream cipher. In: Robshaw, M., Billet, O. (eds.) *New Stream Cipher Designs*. LNCS, vol. 4986, pp. 98–118. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68351-3_9
5. Hell, M., Johansson, T., Maximov, A., Meier, W.: The grain family of stream ciphers. In: Robshaw, M., Billet, O. (eds.) *New Stream Cipher Designs*. LNCS, vol. 4986, pp. 179–190. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68351-3_14
6. Babbage, S., Dodd, M.: The MICKEY stream ciphers. In: Robshaw, M., Billet, O. (eds.) *New Stream Cipher Designs*. LNCS, vol. 4986, pp. 191–209. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68351-3_15
7. Cannière, C.D., Preneel, B.: Trivium. In: Robshaw, M., Billet, O. (eds.) *New Stream Cipher Designs*. LNCS, vol. 4986, pp. 244–266. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68351-3_18
8. Babbage, S., Dodd, M.: The stream cipher MICKEY (version 1). ECRYPT Stream Cipher Project Report 2005/015 (2005). <http://www.ecrypt.eu.org/stream>
9. Hong, J., Kim, W.H.: TMD-Tradeoff and State Entropy Loss Considerations of Streamcipher MICKEY. In: Maitra, S., Madhavan, C., Venkatesan, R. (eds.) *INDOCRYPT 2005*. LNCS, vol. 3797, pp. 169–182. Springer, Heidelberg (2005). https://doi.org/10.1007/11596219_14
10. Tischhauser, E.: Nonsmooth cryptanalysis, with an application to the stream cipher mickey. *J. Math. Cryptol.* **4**(4), 317–348 (2010)
11. Ding, L., Guan, J.: Cryptanalysis of MICKEY family of stream ciphers. *Secur. Commun. Netw.* **6**(8), 936–941 (2013)
12. Helleseht, T., Jansen, C., Kazymyrov, O., Kholosha, A.: State space cryptanalysis of the MICKEY cipher. In: *2013 Workshop on Information Theory and Applications*, pp. 1–10. IEEE Press, New York (2013)
13. Khoo, K., Tan, C.H.: New time-memory-data trade-off attack on the estream finalists and modes of operation of block ciphers. In: *7th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2012)*, pp. 20–21. ACM Press, New York (2013). http://www1.spms.ntu.edu.sg/~kkhoongm/TMD_IEEE_n.pdf
14. Ding, L., Jin, C.H., Guan, J., Qi, C.D.: New treatment of the BSW sampling and its applications to stream ciphers. In: Pointcheval, D., Vergnaud, D. (eds.) *AFRICACRYPT 2014*. LNCS, vol. 8469, pp. 136–146. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-319-06734-6_9
15. Banik, S., Maitra, S.: A differential fault attack on MICKEY 2.0. In: Bertoni, G., Coron, J.-S. (eds.) *CHES 2013*. LNCS, vol. 8086, pp. 215–232. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40349-1_13
16. Karmakar, S., Chowdhury, D.R.: Differential fault analysis of MICKEY-128 2.0. In: *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, pp. 52–59. IEEE Press, New York (2013)
17. Banik, S., Maitra, S., Sarkar, S.: Improved differential fault attack on MICKEY 2.0. *Cryptology ePrint Archive, Report 2013/029*. <http://eprint.iacr.org/2013/029.pdf>

18. Karmakar, S., Chowdhury, D.R.: Differential fault analysis of MICKEY family of stream ciphers. Cryptology ePrint Archive, Report 2014/262. <http://eprint.iacr.org/2014/262.pdf>
19. Liu, J., Gu, D.W., Guo, Z.: Correlation power analysis against stream cipher mickey v2. In: International Conference on Computational Intelligence and Security, pp. 320–324. IEEE Press, New York (2010)
20. Sandeep, S., Rajesh, C.B.: Differential power analysis on FPGA implementation of MICKEY 128. In: 3rd IEEE International Conference on Computer Science and Information Technology, pp. 667–671. IEEE Press, New York (2010)
21. Karmakar, S., Chowdhury, D.R.: Scan-based side channel attack on stream ciphers and its prevention. *J. Cryptographic Eng.* **8**(4), 327–340 (2018)
22. Dunkelman, O., Keller, N.: Treatment of the initial value in time-memory-data trade-off attacks on stream ciphers. *Inf. Process. Lett.* **107**(5), 133–137 (2008)
23. Hellman, M.: A cryptanalytic time-memory trade-off. *IEEE Trans. Inf. Theor.* **26**(4), 401–406 (1980)

Authenticated Key Agreement



A General Construction for Password-Based Authenticated Key Exchange from Witness PRFs

Jiehui Nan^(✉), Mengce Zheng, Zilong Wang, and Honggang Hu^(✉)

Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China

{ustcnjh,mczheng,wz10830}@mail.ustc.edu.cn, hghu2005@ustc.edu.cn

Abstract. In cyber security, authenticated key exchange (AKE) can be used to achieve the privacy and authentication of data. As a relevant cryptographic protocol, password-based authenticated key exchange (PAKE) has been studied for its convenience. Recently, Katz and Vaikuntanathan proposed a round-optimal PAKE from smooth projective hash functions (SPHFs). However, the instantiation of smooth projective hash functions depends on the underlying NP-relation which is a CCA-secure encryption relation in their construction. In this paper, we apply a new cryptographic primitive named witness PRFs to construct PAKE. In our settings, the concrete construction of witness PRFs is independent of the underlying NP-relation. At this point, our construction is more general, and furthermore, we have a discussion on some possible NP-relations, which could be used to construct secure PAKE in our settings.

Keywords: Authenticated key exchange · Witness PRFs · CCA-secure labeled encryption · OAEP+

1 Introduction

Nowadays, people can conveniently communicate with each other via the development of the information technology and the popularity of the intelligent terminals. Therefore, the data security is more and more concerned by the terminal users. The main tool to achieve the secure goal is the cryptography, and to communicate secretly and reliably for different parties within a malicious environment is an important problem in cryptography, which can be achieved by generating a secret session key between the parties through a protocol. Then, using the session key, parties can apply symmetric encryption and MACs to communicate securely with each other.

The problem of generating a secret session key was first studied by Diffie and Hellman [20] which shows how the two parties can share a session key securely against a passive adversary who can eavesdrop on the communication data.

Unfortunately, the Diffie-Hellman key exchange protocol can not resist against the man-in-the-middle attack since it does not provide any form of authentication. Inherently, it requires some information to be shared between the parties to achieve the authentication. Some two-party authentication key exchange (AKE) protocols have been designed [3,6,21], but they all need the two parties to share a high-entropy information. Subsequently, some researchers find that a min-entropy shared information is enough for AKE, and the main protocol is password-based authentication key exchange (PAKE).

Informally, PAKE protocol enables the two parties to generate a session key just using a simple shared password which is chosen from a small set of all possible values. In this setting, the password only maintains a min-entropy, so that it can bring some advantages: cheap and human-memorable. However, it also incurs the defect that the low-entropy password may be easily discovered by brute force. Assuming that an adversary gets a password-dependent data, it can guess and check the password using this data. Such attack is called dictionary attack. When discussing a secure protocol, two types of attack should be considered. The first one is off-line attack and in this case, the adversary eavesdrops on the transmitted messages between the two parties, then tries password testing privately. The second attack type is online attack, which means that the adversary actively involves in the protocol interaction and tries to acquire the session key or even guess the correct password. It should be sure that PAKE must be able to resist against off-line attack. On the other hand, it is known that on-line attack is unavoidable, but this problem can be solved by restricting the number of the online queries.

The seminal work of PAKE was given by Bellare and Merrit [4], and the formal security models were proposed by Bellare et al. [9] and Boyko et al. [10]. After that, a large number of constructions were proposed under the random oracle model [1,9,10]. The first PAKE protocol in standard model was given by Goldreich and Lindell [14], which was improved and simplified subsequently, but still inefficient. The first efficient PAKE under DDH assumption was demonstrated by Katz et al. [17], which needs a common reference string (CRS) setting. Gennaro and Lindell [14] presented a framework of PAKE based on the results of Katz et al. [17], and this framework consists of two smooth projective hash functions (SPHF) and a CCA-secure encryption scheme. Recently, Katz and Vaikuntanathan [18] use SPHF and CCA-secure labeled encryption to construct a round-optimal PAKE scheme, which only needs one-round complexity to achieve the implicit authentication. SPHF play an important role in construction of PAKE. However, when using SPHF to construct PAKE, it needs to instantiate the SPHF based on the underlying encryption scheme. Fortunately, there is a similar cryptographic primitive named witness pseudo-random functions (witness PRFs) introduced by Zhandry in [25]. The property of witness PRFs is similar to SPHF, but suitable for any NP-relation (including encryption-relation or commitment-relation etc.), which is also equivalent to the definition of NP-Language. Note that one application of witness PRFs [25] is to build witness encryption (WE) for any NP-language and one related work

mentioned in [13] is to construct WE from SPHF's for some special algebraic languages.

In this paper, our main contribution is applying witness PRFs to construct PAKE. In our new settings, the concrete construction of witness PRFs is independent of the underlying NP-relation, and we mainly consider the CCA-secure labeled encryption relation later. This property enables us to instantiate the underlying NP-relation by different ways without considering the construction of witness PRFs. To reveal this advantage, we follow the framework of [18] and construct a new CCA-secure labeled encryption scheme based on OAEP+, which is suitable for any one-way trapdoor permutation instead of other schemes based on DDH. Furthermore, we discuss, to achieve a secure PAKE protocol, what structure of the underlying NP-relation should be, and this discussion is inspired by the work of Xue et al. [22]. Lastly, we propose an open problem for replacing the one-way trapdoor permutation by the one-way permutation in our new construction of CCA-secure labeled encryption scheme based on OAEP+.

The rest of this paper is organized as follows. In Sect. 2, we introduce some definitions and notations, and also present the security model of PAKE. In Sect. 3, we present the protocol of PAKE and give a brief security proof. In Sect. 4, we construct a CCA-secure labeled encryption scheme based on OAEP+. We give a further discussion in Sect. 5. Finally, conclusion is given in Sect. 6.

2 Preliminaries

2.1 Definitions and Notations

Throughout this paper, we denote \mathbb{N}^+ as the positive integer set and $\lambda \in \mathbb{N}^+$ as the security parameter. We say a function $f : \mathbb{N} \rightarrow \mathbb{R}$ negligible if $\forall c \exists n_c$ such that if $n > n_c$ then $f(n) < n^{-c}$, and we use $\varepsilon(n)$ to denote the corresponding negligible function of n in this paper. We use “:=” for deterministic assignment, “ \leftarrow ” for randomized assignment, and “=” for equivalence. We use U_k to denote the uniform distribution over $\{0, 1\}^k$ and we specially denote the randomly sampling set as $\{0, 1\}^{n_R}$ for the random encryption scheme used in our settings.

Witness PRFs were proposed by Zhandry in [25], and one construction is based on multilinear maps. This cryptographic primitive is similar to SPHF's, but the construction of SPHF's is feasible only for certain languages, such as certain group-theoretic languages. In contrast, witness PRFs can handle arbitrary NP-languages and such flexibility is required in this paper. Now we present the definition of a witness PRF as follows:

Definition 1 (Witness PRF [25]). *A witness PRF is a tuple of algorithms (Gen, F, Eval) such that:*

1. Gen is a randomized algorithm that takes as input a security parameter λ , a relation-circuit $R: \mathcal{X} \times \mathcal{W} \rightarrow \{0, 1\}$, and randomly generates a secret key fk and a public evaluation key ek .

2. F is a deterministic algorithm that takes as input the secret key fk and an input $x \in \mathcal{X}$, then computes $y := F(fk, x) \in \mathcal{Y}$ as the output for some set \mathcal{Y} .
3. Eval is a deterministic algorithm that takes as input the public evaluation key ek , an input $x \in \mathcal{X}$, and a witness $w \in \mathcal{W}$, then computes $y := \text{Eval}(ek, x, w)$ as the output, where $y \in \mathcal{Y}$.

We define a NP-language for the relation-circuit R as

$$L_R = \{x \in \mathcal{X} \mid \exists w \in \mathcal{W} \text{ s.t. } R(x, w) = 1\}.$$

We informally describe the correctness and adaptive security as follows: for all $x \in L_R$, if w is a witness i.e. $R(x, w) = 1$, then $\text{Eval}(ek, x, w) = F(fk, x)$, otherwise, $\text{Eval}(ek, x, w)$ is an independent value to $F(fk, x)$. For all $x \notin L_R$, $F(fk, x)$ is computationally distinguishable from a random value even given ek and we say the witness PRFs adaptively-secure witness PRFs if it allows the adversary to query adaptively for polynomial times. It should be noted here that in any formal definition, the algorithm Eval outputs a termination symbol \perp in the situation like $R(x, w) = 0$. The formal security definition and construction of witness PRFs can be found in [25].

For describing the PAKE protocol conveniently, here we set $\mathcal{X} = \{0, 1\}^{n_{F_{in}}}$ and $\mathcal{Y} = \{0, 1\}^{n_{F_{out}}}$, where $n_{F_{in}}, n_{F_{out}} \in \mathbb{N}^+$. Next we give the definition of labeled public-key encryption scheme and the corresponding CCA-secure model.

Definition 2 ([18]). *A labeled public-key encryption scheme is a tuple of PPT algorithms $(\text{Gen}, \text{Enc}, \text{Dec})$ such that:*

1. Gen is a randomized algorithm that takes as input a security parameter λ and returns a public key pk and a secret key sk .
2. Enc is a randomized algorithm that takes as inputs a public key pk , a label Label , and a message m and returns a ciphertext $C := \text{Enc}_{pk}(\text{Label}, m; r)$ where r is chosen randomly from $\{0, 1\}^{n_R}$.
3. Dec is a deterministic algorithm that takes as inputs a secret key sk , a label Label , and a ciphertext C and returns a message m or a symbol \perp . We write this as $m := \text{Dec}_{sk}(\text{Label}, C)$.

In order to avoid confusions, we should distinguish two presentations as follows: $C \leftarrow \text{Enc}_{pk}(\text{Label}, m)$ means that C is a randomized assignment from the encryption space of (Label, m) , where the encryption result is based on the chosen of random strings. Otherwise, $C := \text{Enc}_{pk}(\text{Label}, m; r)$ means that the random string r is fixed and C can be computed deterministically.

To model the security for the labeled encryption scheme, it needs to define a left-or-right encryption oracle $\text{Enc}_{pk,b}(\cdot, \cdot, \cdot)$, where $b \in \{0, 1\}$, as follows:

$$\text{Enc}_{pk,b}(\text{Label}, m_0, m_1) \stackrel{\text{def}}{=} \text{Enc}_{pk}(\text{Label}, m_b).$$

Definition 3 ([18]). *A labeled public-key encryption scheme $(\text{Gen}, \text{Enc}, \text{Dec})$ is CCA-secure if for all PPT adversary, the advantage defined in following is negligible:*

$$|2 \cdot \Pr[(pk, sk) \leftarrow \text{Gen}(1^n); b \leftarrow \{0, 1\} : \mathcal{A}^{\text{Enc}_{pk,b}(\cdot, \cdot), \text{Dec}_{sk}(\cdot, \cdot)}(1^n, pk) = b] - 1|,$$

where \mathcal{A} 's queries are restricted to that, for any query $\text{Enc}_{pk,b}(\text{Label}, m_0, m_1)$, there should be $|m_0| = |m_1|$ i.e. the two underlying messages have the same length. Another restriction is that \mathcal{A} can not query the decryption oracle for (Label, C) where C is the challenge response encrypted with Label (but it is allowed to query $\text{Dec}_{sk}(\text{Label}', C)$ with $\text{Label}' \neq \text{Label}$).

2.2 Security Model of PAKE

The security model for PAKE was firstly proposed by Bellare et al. [9], and their main contribution was based on prior works of [5, 8]. This security model also was used in [17, 18], and we follow the notations and descriptions from them. The details are described below.

The protocol is based on common reference string (CRS), which was set as one-way trapdoor permutation in our scheme. We denote a set which includes all protocol participants by \mathcal{U}_{ser} . We treat any user $U \in \mathcal{U}_{ser}$ as a bit string which presents the identity of the user. For two different users $U, U' \in \mathcal{U}_{ser}$, we denote $pw_{UU'}$ as their shared password. The passwords we used in practice are all bit strings (i.e. the passwords space is $\{0, 1\}^{n_p}$). However, in concrete discussion, it should distinguish the valid and invalid passwords. Without loss of generality, we denote the valid password set by \mathcal{V}_{pw} , the invalid password set $\bar{\mathcal{V}}_{pw} := \{0, 1\}^{n_p} \setminus \mathcal{V}_{pw}$ (we can transform the notations discussed in [18] into our definitions, just using some normal encoding). In our new settings, arbitrary distributions of valid passwords are feasible.

Before any execution of protocol, it needs an initialization step. In the formal model, the adversary can send messages to and receive messages from any users by impersonating another user. Also the adversary can eavesdrop on the messages passively from the correct execution between two valid users. In fact, any users can execute the protocol concurrently with the others. We let Π_U^i denote instance i of user U as an oracle and each instance may be executed only once. We treat the adversary \mathcal{A} as a probabilistic polynomial algorithm which can access any oracle like Π_U^i at any time. Each instance Π_U^i maintains states, and we denote the corresponding variables as follows:

- \mathbf{sid}_U^i , \mathbf{pid}_U^i , and \mathbf{sk}_U^i denotes the session id, partner id, and session key for an instance Π_U^i . The session id is an identifier which can be used to keep track of different executions and \mathbf{sid}_U^i consists of the concatenation of all messages sent and received by Π_U^i . The partner id represents a user, with which the instance believes to generate a session key via the PAKE protocol.
- \mathbf{acc}_U^i and \mathbf{term}_U^i are boolean variables, denoting whether a given instance has accepted or terminated.

The adversary is assumed to have complete controls over all communications in the network. We model the adversary’s query strategy via accessing the oracles as follows:

- **Send**(U, i, M). This oracle query sends messages M to instance Π_U^i . The instance then computes as protocol says, and sends back the response. At the same time, Π_U^i updates the states and the adversary can get the messages output by Π_U^i .
- **Execute**(U, i, U', j). Assuming that instance Π_U^i and $\Pi_{U'}^j$ have not been used, this oracle executes the protocol between these two instances. An adversary can collect all the transcripts of the protocol execution after this query.
- **Reveal**(U, i). If instance Π_U^i has accepted and holden a session key sk_U^i . This oracle outputs the session key. In fact, this oracle models the leakage of the session key for a variety of reasons, including hacking or cryptanalysis.
- **Test**(U, i). This oracle query is used to define the security of protocol like the distinguishable definition for semantic security. Concretely, the oracle randomly chooses a bit b ; if $b = 1$, then the adversary is given sk_U^i , and if $b = 0$, the adversary is given a session key chosen uniformly from a appropriate space. The adversary’s advantage is to distinguish these two cases.

Let $U, U' \in \mathcal{U}_{ser}$. We define that instances Π_U^i and $\Pi_{U'}^j$ are partnered if $sid_U^i = sid_{U'}^j, \neq NULL, pid_U^i = U'$ and $pid_{U'}^j = U$. We define the correctness of the protocol execution as follows: if Π_U^i and $\Pi_{U'}^j$ are partnered, then $acc_U^i = acc_{U'}^j = TRUE$ and $sk_U^i = sk_{U'}^j$.

Now, we discuss the adversary’s advantage. To define the adversary’s success, we first introduce a notion of freshness. An instance Π_U^i is fresh if the adversary does not get this instance’s session key through the trivial query (i.e. just using a **Reveal** query on that instance). and the **Test** query is only confined to a fresh instance when we consider the adversary’s success advantage. After a series of **Send**, **Execute**, and **Reveal** queries, the adversary finally makes a single query **Test**(U, i) to a fresh instance Π_U^i , and outputs a bit b' . We denote the event $b' = b$ by **Succ**. Finally, the advantage of the adversary \mathcal{A} is defined as $Adv_{\mathcal{A}, \Pi}(n) = 2Pr[\mathbf{Succ}] - 1$.

It is obvious that a probabilistic polynomial time (PPT) adversary can always succeed with probability 1 by trying all passwords one-by-one efficiently, since the size of the password set is small. Informally, we say a PAKE protocol is secure against on-line attack if to enumerate the password is the best way, when attacking the protocol. Finally, we define the secure PAKE protocol against on-line attack as follows:

Definition 4 ([18]). *The Protocol Π is a secure protocol for password-based authenticated key exchange (PAKE) if, for all PPT adversary \mathcal{A} , making at most $Q(n)$ on-line attacks, it holds that $Adv_{\mathcal{A}, \Pi}(n) \leq Q(n)/N + \varepsilon(n)$, where N is the size of the valid password set.*

3 One-Round Secure PAKE Protocols

In this section, we construct a PAKE protocol using witness PRFs and the corresponding NP-language. To show the advantage of our construction, the NP-language we used here is converted from the CCA-secure labeled public-key encryption scheme. In our settings, it doesn't need any other restrictions in contrast with [17, 18].

Here, the public parameter is the public key pk generated by $\text{Gen}(1^n)$ (In fact, pk is derived from a common random string). We use the framework of OAEP+ to construct the CCA-secure labeled public-key encryption scheme later, and therefore, the public key is in fact the one-way trapdoor permutation f . Now, we define the NP-language as $L = \{(Label, C, m) \in \{0, 1\}^{poly(n)} \mid \exists r \text{ s.t. } C = \text{Enc}_{pk}(Label, m; r) \wedge m \in \mathcal{V}_{pw}\}$, and the corresponding NP-relation is denoted by R_L .

Consider the execution of the protocol between two different users U and U' , which share a same password $pw := pw_{UU'}$. We describe the execution on the view of U below; see also Fig. 1.

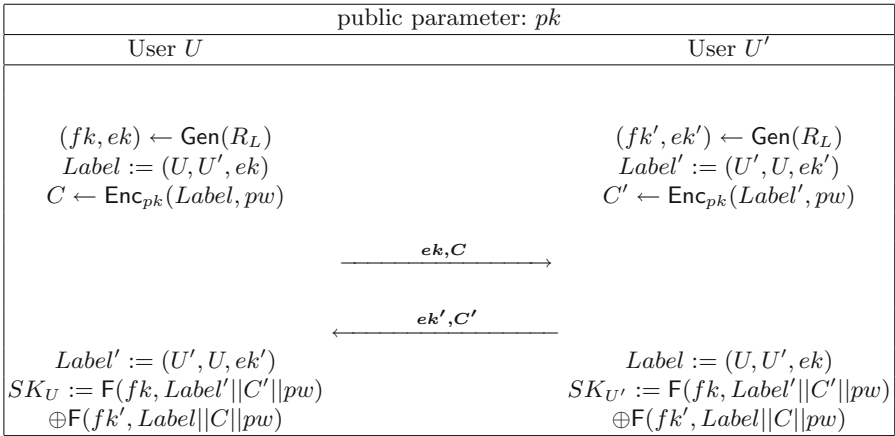


Fig. 1. Our PAKE protocol with witness PRFs

First, user U randomly generates the key pairs based on the NP-relation i.e. $(fk, ek) \leftarrow \text{Gen}(R_L)$, then it sets $Label := (U, U', ek)$ and computes the ciphertext $C := \text{Enc}_{pk}(Label, pw; r)$ using the local random string r , which can also be expressed as $C \leftarrow \text{Enc}_{pk}(Label, pw)$. The user U sends the message (ek, C) to U' , and the user U' computes and sends message (ek', C') similarly. If (ek', C') is not a valid message, then U simply rejects. Otherwise, U sets $Label' := (U', U, ek')$ and $sk_U := F(fk, Label' || C' || pw) \oplus F(fk', Label || C || pw)$, where U can compute $F(fk, Label' || C' || pw)$ just using its secret key fk , and can compute $F(fk', Label || C || pw)$ using ek' and the witness r corresponding to

$x := (Label, C, pw) \in L$. The correctness is easy to see, and next, we analyse the security of our protocol.

Theorem 1 (Security). *If $(Gen, F, Eval)$ is a adaptively-secure witness pseudorandom function, and the underlying NP-relation is a CCA-secure labeled encryption relation, then the protocol in Fig. 1 is a secure PAKE protocol.*

Proof. (sketch) The proof proceeds by a sequence of games. Concretely, for any probability polynomial time (PPT) adversary \mathcal{A} attacking the protocol Π , we construct six games **Game 0**, \dots , **Game 5**, with the original protocol corresponding to **Game 0**. Let $Adv_{\mathcal{A},i}(n)$ denote the advantage of \mathcal{A} against the **Game i**. The key of proof is to bound on $Adv_{\mathcal{A},\Pi}(n) = Adv_{\mathcal{A},0}(n)$. Next, we show that any gap between advantages of two successive games is negligible and the advantage of the last game can be bounded easily from the information-theoretic view, so that we can conclude the result via the hybrid argument.

Game 1: We change the **Execute** query. Specially, instead of encrypting the password pw , we encrypt an invalid password pw_{\perp} . Then the user U and U' compute $C \leftarrow Enc_{pk}(Label, pw_{\perp})$ and $C' \leftarrow Enc_{pk}(Label', pw_{\perp})$ respectively in this game. The session key is computed as $sk'_{U'} := sk_U := F(fk, Label' || C' || pw) \oplus F(fk', Label || C || pw)$ where the witness PRF can be computed using the secret keys fk and fk' , and pw is the password shared between the two users.

Claim 1. $|Adv_{\mathcal{A},0}(n) - Adv_{\mathcal{A},1}(n)|$ is negligible.

Proof. The claim is correct immediately from semantic security of the encryption scheme. Otherwise, if $|Adv_{\mathcal{A},0}(n) - Adv_{\mathcal{A},1}(n)|$ is not negligible, then there is an efficient algorithm, which can use this gap to break down the semantic security of the underlying encryption scheme.

Game 2: In this game, we again change the **Execute** query. This game is same as the first game except that the session key $sk_U = sk_{U'}$ is chosen uniformly from $\{0, 1\}^{n_{F_{out}}}$.

Claim 2. $|Adv_{\mathcal{A},1}(n) - Adv_{\mathcal{A},2}(n)|$ is negligible.

Proof. This claim is correct due to the property of witness PRFs. When the **Execute** was queried, the messages given to the adversary are (ek, C, ek', C') with $C \leftarrow Enc_{pk}(Label, pw_{\perp})$ and $C' \leftarrow Enc_{pk}(Label', pw_{\perp})$. In **Game 1**, the session key is computed as $sk'_{U'} = sk_U := F(fk, Label' || C' || pw) \oplus F(fk', Label || C || pw)$, where pw is the password shared between U and U' . Since $(Label', C', pw) \notin L$, we conclude that $(ek, F(fk, Label' || C' || pw))$ and $(ek, U_{n_{F_{out}}})$ are computationally indistinguishable based on the property of the witness PRFs. This means that session keys of the **Game 1** and **Game 2** are computationally indistinguishable even given the transcripts from the execution of the protocol, and the claim follows.

We treat the **Send** query as two different types. The **Send0**(U, i, U') query causes instance Π_U^i to initiate the protocol with the user U' . After this query, the adversary gets messages sent from U to U' , and at the same time the state updates with $pid_U^i := U'$. The second type of **Send** query is **Send1**(U, i, \mathbf{msg}), which represents \mathcal{A} sends the message \mathbf{msg} to instance Π_U^i . The affection of this query is that the session key can be computed, which will affect the subsequent **Reveal** or **Test** query for instance Π_U^i . Here, it is convenient to ignore the invalid message case. We assume pid_U^i of **Send1**(U, i, \mathbf{msg}) is U' . A valid message \mathbf{msg} is said previously used if it was output by a previous **Send0**($U, *, U'$) query, and otherwise is said adversarially generated.

Game 3: In this game, The simulator not only generates the public parameter pk , but also stores the associated secret key sk , and this subtle modification is a syntactic change. Another main modification against **Game 2** is the **Send1** query. Specially, we consider the **Send1**(U, i, \mathbf{msg}) query where $\mathbf{msg} := (ek', C')$. We note that $pid_U^i := U'$, $Label' := (U', U, ek')$, and pw is the password shared between U and U' . We show three cases to response the query as follows:

1. If the message \mathbf{msg} is adversarially generated, then the simulator computes $pw' := Dec_{sk}(Label', C')$. If $pw' = pw$, the simulator declares that adversary wins the game.
2. If the message \mathbf{msg} is adversarially generated, then the simulator computes $pw' := Dec_{sk}(Label', C')$. If $pw' \neq pw$, the simulator chooses sk_U^i uniformly from $\{0, 1\}^{n_{F_{out}}}$.
3. If the message \mathbf{msg} is previously used, then the simulator computes the session key $sk_U := F(fk, Label' || C' || pw) \oplus F(fk', Label || C || pw)$ using fk and fk' .

Claim 3. $Adv_{\mathcal{A},2}(n) \leq Adv_{\mathcal{A},3}(n) + \varepsilon(n)$ where $\varepsilon(n)$ is a negligible function of n .

Proof. We now analyse the three cases above. It's obvious that the change in first case just only increases the advantage of \mathcal{A} , and the claim is trivial. In case 2, we have $(Label', C', pw) \notin L$ since $pw' \neq pw$. We conclude that $(ek, F(fk, Label' || C' || pw))$ and $(ek, U_{n_{F_{out}}})$ are computationally indistinguishable due to the property of witness PRFs, and the claim is correct in this case. In the last case, the computation of session key does not be affected since $(Label, C, pw) \in L$.

Game 4: We change the **Send1** query once more. If **Send1**(U, i, \mathbf{msg}) is previously used, where $\mathbf{msg} := (ek', C')$ and $pid_U^i := U'$, then the simulator proceeds for two opposite cases as follows:

1. There exists an instance Π_U^j , partnered with Π_U^i i.e. the view of their transcripts is same. In this case, the simulator sets $sk_U^i := sk_U^j$.
2. There does't exist any instance partnered with Π_U^i . In this case, the simulator chooses sk_U^i uniformly from $\{0, 1\}^{n_{F_{out}}}$.

Claim 4. $|Adv_{\mathcal{A},3}(n) - Adv_{\mathcal{A},4}(n)|$ is negligible.

To prove the Claim 4, we first need to prove a relative lemma based on the following experiment **Expt_b** where $b \in \{0, 1\}$:

1. Compute $(pk, sk) \leftarrow \text{Gen}(1^n)$ as the public and secret key for the CCA-secure labeled encryption scheme. Let $(\text{Gen}, \text{F}, \text{Eval})$ be a adaptively-secure witness PRF and the corresponding NP-language L is based on the CCA-secure labeled encryption scheme with the valid password space. Give pk to the adversary \mathcal{A} .
2. Formally, denote that $l = l(n)$. And randomly generate l key pairs $(fk_1, ek_1), \dots, (fk_l, ek_l) \leftarrow \text{Gen}(R_L)$. Given ek_1, \dots, ek_l to \mathcal{A} .
3. \mathcal{A} may adaptively query a labeled encryption oracle that takes input $(Label, pw)$ where pw is a valid password, and outputs a ciphertext $C \leftarrow \text{Enc}_{pk}(Label, pw)$ along with:
 - a. If $b = 0$, the values $F(fk_i, Label || C || pw)$ for $i = 1$ to l .
 - b. If $b = 1$, the independently random values r_1, \dots, r_l chosen from $\{0, 1\}^{n_{F_{out}}}$.
4. \mathcal{A} can adaptively query a decryption oracle $\text{Dec}_{sk}(\cdot, \cdot)$ at any point except the pair $(Label, C)$, where C was obtained from the encryption oracle in 3.
5. Finally, \mathcal{A} outputs a bit b' and we say \mathcal{A} succeeds if $b' = b$.

Lemma 1. Let $(\text{Gen}, \text{Enc}, \text{Dec})$ be a CCA-secure labeled encryption scheme, and $(\text{Gen}, \text{F}, \text{Eval})$ a adaptively-secure witness PRF, for any PPT adversary \mathcal{A} , we have $Pr[\mathcal{A} \text{ succeeds}] \leq \frac{1}{2} + \varepsilon(n)$, where $\varepsilon(n)$ is a negligible function of n .

The proof of Lemma 1 is similar to that of Lemma 1 in [18]. Here we only show the main idea of the proof. The results of encrypting a valid password and an invalid password are computationally indistinguishable since the labeled encryption scheme is semantic secure. The encryption of an invalid password will get a instance which not belongs to the corresponding NP-language; in this settings, the value of witness PRF for the instance is computationally indistinguishable from the random string even though the decryption oracle has been queried for polynomial times due to its CCA-secure property. After polynomial encryption oracle queries, **Expt₀** and **Expt₁** are still computationally indistinguishable via a hybrid argument.

Proof of Claim 4: Assume \mathcal{A} queries the **Send** $l = \text{poly}(n)$ times. Now we construct a simulator \mathcal{S} interacting with the experiment in Lemma 1 as follows:

1. \mathcal{S} gets the public parameter pk and ek_1, \dots, ek_l samely as the adversary in the experiment defined in Lemma 1. Then \mathcal{S} sends the public information to \mathcal{A} .
2. \mathcal{S} chooses the password $pw_{UU'}$ randomly for all different users U and U' .
3. \mathcal{S} responses to **Execute** query samely as the simulator in **Game 2**.
4. \mathcal{S} responses to the i th **Send0** query **Send0** $(U, *, U')$ as follows: sets $Label := (U, U', ek_i)$ and sents $(Label, pw_{UU'})$ to the encryption oracle defined in Lemma 1. After receiving the ciphertext C_i along with $f_{1,i}, \dots, f_{l,i}$, \mathcal{S} gives (ek_i, C_i) to \mathcal{A} .

5. \mathcal{S} responds to **Send1**(U, j, \mathbf{msg}) query where $\mathbf{msg} := (ek', C')$ as follows: if there exists an instance $\Pi_{U'}^k$, partnered with $\Pi_{U'}^j$, then, it sets $sk_{U'}^j := sk_{U'}^k$. Otherwise, lets $pid_{U'}^j := U'$ and $Label' := (U', U, ek')$, and in this case, assuming **Send0**(U, j, U') query was the i th **Send0** query with the transcript (ek_i, C_i) , \mathcal{S} responds in two different ways:
 - a. If \mathbf{msg} is previously used, and it was output by the r th **Send0** query **Send0**($U', *, U$) with the transcript (ek_r, C_r) . Then, \mathcal{S} computes $sk_{U'}^j := f_{i,r} \oplus f_{r,i}$
 - b. If \mathbf{msg} is adversarially generated, then \mathcal{S} sends $(Label', C)$ to the decryption oracle and receives a value pw' . Samely as **Game3**, if $pw' = pw_{UU'}$, then \mathcal{S} declares that \mathcal{A} succeeds and terminates, and if $pw' \neq pw_{UU'}$, then $sk_{U'}^j$ is chosen randomly from $\{0, 1\}^{n_{F_{out}}}$.
6. \mathcal{S} outputs 1 if and only if \mathcal{A} succeeds.

Next, we show that the simulator \mathcal{S} bridges the view of \mathcal{A} against the games to the view of \mathcal{A} in above execution. Let b be same as it in the experiment defined above. If $b = 0$, in step 5(a), it holds that $f_{i,r} = F(fk_i, Label' || C_r || pw_{UU'})$ and $f_{r,i} = F(fk_r, Label || C_i || pw_{UU'})$. The view of \mathcal{A} in this case is identical to the view of \mathcal{A} in **Game3**. If $b = 1$, all the values $f_{i,j}$ received by \mathcal{S} are independently random strings. It holds that all session keys computed in step 5(a) are uniformly and independently since $f_{i,r}$ is random and used only once in this step. It's not hard to see that the view of \mathcal{A} in this case is identical to the view of \mathcal{A} in **Game4**. The Claim 4 follows from the Lemma 1.

Game5: This game is same to **Game4** except we only change the behave of the **Send0** query. Specially, we response (ek, C) to **Send0**(U, i, U') query, where C is no longer the encryption of $pw_{UU'}$ but an encryption of an invalid password.

Claim 5. $|Adv_{\mathcal{A},4}(n) - Adv_{\mathcal{A},5}(n)|$ is negligible.

Proof. The proof of this claim is immediately from that (Gen, Enc, Dec) is a CCA-secure labeled encryption scheme.

Now, we analyse the bound of any PPT adversary's advantage against **Game5**, where the view of \mathcal{A} is independent of any user's passwords. After $Q(n)$ on-line queries, the only advantage of \mathcal{A} is submitting an adversarially generated message that is corresponding to an encryption of a correct password. And it bounds that $Adv_{\mathcal{A},5}(n) \leq Q(n)/N$, where $N = |\mathcal{V}_{pw}|$. Combining with claims 1–5, we can conclude that $Adv_{\mathcal{A},5}(n) \leq Q(n)/N + \varepsilon(n)$ where $\varepsilon(n)$ is a negligible function of n . That proves the Theorem 1.

4 A Construction of Labeled CCA-secure Encryption from OAEP+

In cryptography, Optimal Asymmetric Encryption Padding (OAEP) is a padding scheme, often used together with RSA encryption, and it was first introduced by

Bellare and Rogaway in [7]. However, this scheme was proved to be CCA-secure only when the RSA permutation is used. Subsequently, an improved scheme called OAEP+ that can work with any one-way trapdoor permutation, was proposed by Victor Shoup in [19]. Here we omit the scheme of OAEP+, and directly construct the CCA-secure labeled encryption scheme based on OAEP+.

Before describing the encryption scheme, we first define parameters k, n, k_0, k_1 that satisfy $k_0 + k_1 < k$ and $n = k - k_0 - k_1$. The scheme also needs three functions which are modeled as independent random oracles:

$$\begin{aligned} G &: \{0, 1\}^{k_0} \rightarrow \{0, 1\}^n, \\ H' &: \{0, 1\}^{n+k_0} \rightarrow \{0, 1\}^{k_1}, \\ H &: \{0, 1\}^{n+k_1} \rightarrow \{0, 1\}^{k_0}. \end{aligned}$$

Now, we describe the key generation, encryption, and decryption algorithm of the scheme in following:

Key Generation. First, the key generation algorithm generates a one-way trapdoor permutation f and a corresponding trapdoor g randomly. Then, it sets f as the public key, and g as the private key.

Encryption. Given a plaintext $m \in \{0, 1\}^n$ and a label $Label \in \{0, 1\}^{k_1}$, the encryption algorithm randomly chooses $r \in \{0, 1\}^{k_0}$ and computes:

$$\begin{aligned} s &:= G(r) \oplus m || H'(r || m) \oplus Label, \\ t &:= H(s) \oplus r, \\ w &:= s || t, \\ C &:= f(w). \end{aligned}$$

where

$$s \in \{0, 1\}^{n+k_1}, t \in \{0, 1\}^{k_0}, w \in \{0, 1\}^k, C \in \{0, 1\}^k.$$

The encryption algorithm outputs $(Label, C)$.

Decryption. Given a label-ciphertext pair $(Label, C)$, the decryption algorithm computes

$$\begin{aligned} w &\in \{0, 1\}^k, s \in \{0, 1\}^{n+k_1}, t \in \{0, 1\}^{k_0}, \\ r &\in \{0, 1\}^{k_0}, m \in \{0, 1\}^n, Pad \in \{0, 1\}^{k_1}, \end{aligned}$$

as follows:

$$\begin{aligned} w &:= g(C), \\ s &:= w[0\dots n + k_1 - 1], \\ t &:= w[n + k_1\dots k], \\ r &:= H(s) \oplus t, \\ m &:= G(r) \oplus s[0\dots n - 1], \\ Pad &:= s[n\dots n + k_1 - 1]. \end{aligned}$$

If $Pad \oplus Label = H'(r || m)$, then the algorithm outputs the cleartext m . Otherwise, it outputs a symbol \perp . The security proof of our scheme follows [19].

5 Further Discussion

In this section, we first discuss what structure of the underlying NP-language should be in our settings. In fact, Xue et al. [22] have showed that plaintext checkable attack secure (PCA-secure) key encapsulation mechanism (KEM) is enough for constructing the PAKE protocol rather than CCA-secure encryption. Instead, we can just use the PCA-secure encryption scheme in our settings. Now, we informally define the PCA-secure labeled encryption scheme. Like the CCA-secure labeled encryption scheme, PCA-secure labeled encryption scheme also has a tuple of PPT algorithms (Gen, Enc, Dec) which have the same definitions as the CCA case, but its security model has changed. Instead of the Decryption oracle query, we consider the **Dcheck** oracle query here. Any PPT adversary can query the instance $(Label, C, m)$ to the **Dcheck** oracle, and the oracle responses $b := 1$, if C is a valid ciphertext for $Label$ and m (i.e. $C \leftarrow Enc_{pk}(Label, m)$), otherwise, the oracle responses $b := 0$. Here, we can convert the PCA-secure labeled encryption scheme to a NP-language instance which can be used to construct PAKE along with adaptively-secure witness PRFs. It is easy to see that the PCA-secure encryption implies the semantic secure encryption and is weaker than CCA-secure encryption, but its security property is enough in our settings. In fact, essentially, both the CCA-secure encryption and the PCA-secure encryption have a same property called non-malleability. Another useful NP-language instance is the non-malleability commitment scheme, which was used to construct PAKE by Gennaro and Lindell in [15].

Another interesting problem we need to discuss is replacing the one-way trapdoor permutation in our CCA-secure labeled encryption scheme by the one-way permutation. In fact, only the encryption algorithm is used in our PAKE protocol scheme. Concretely, consider our new instantiation of CCA-secure labeled encryption scheme, its trapdoor is not used in the PAKE scheme. Furthermore, the inherent property needed in our settings is the non-malleability of encryption scheme, and this property is still holding for the one-way permutation instead of the one-way trapdoor permutation. The only difference here is that this change does not have an efficient decryption algorithm which can be used to prove the security of our scheme. For this reason, we let the security proof of the changed scheme replacing the one-way trapdoor permutation by the one-way permutation be an open problem.

6 Conclusion

In this paper, we construct PAKE protocol by using witness PRFs. Compared with other schemes, the concrete construction of witness PRFs in our settings is independent of the underlying NP-relation like CCA-secure labeled encryption-relation. For this reason, we can instantiate the underlying NP-relation by different ways without considering the construction of witness PRFs. To reveal the advantage of our new settings, we construct a new CCA-secure labeled encryption scheme based on OAEP+, which can be instantiated with any one-way

trapdoor permutation, instead of other encryption schemes based on DDH. Furthermore, we propose an open problem about proving the security of the changed scheme replacing the one-way trapdoor permutation by the one-way permutation. Additionally, we have a discussion on some possible NP-relations which can be used to construct secure PAKE protocol.

Acknowledgment. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was partially supported by the National Natural Science Foundation of China (Grant No. 61632013).

References

1. Abdalla, M., Pointcheval, D.: Simple password-based encrypted key exchange protocols. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 191–208. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30574-3_14
2. Abdalla, M.: Password-based authenticated key exchange: an overview. In: Chow, S.S.M., Liu, J.K., Hui, L.C.K., Yiu, S.M. (eds.) ProvSec 2014. LNCS, vol. 8782, pp. 1–9. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12475-9_1
3. Bird, R., et al.: The kryptoknight family of light-weight protocols for authentication and key distribution. IEEE/ACM Trans. Networking **3**(1), 31–41 (1995)
4. Bellare, S.M., Merritt, M.: Encrypted key exchange: password-based protocols secure against dictionary attacks. In: IEEE S&P, pp. 72–84 (1992)
5. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48329-2_21
6. Bellare, M., Canetti, R., Krawczyk, H.: A modular approach to the design and analysis of authentication and key exchange protocols. In: 30th Annual ACM Symposium on Theory of Computing (STOC 1998), pp. 419–428. ACM, Dallas (1998)
7. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995). <https://doi.org/10.1007/BFb0053428>
8. Bellare, M., Canetti, R., Krawczyk, H.: Provably secure session key distribution: the three party case. In: 27th Annual ACM Symposium on Theory of Computing (STOC 1995), pp. 57–66. ACM, Las Vegas (1995)
9. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_11
10. Boyko, V., MacKenzie, P., Patel, S.: Provably secure password-authenticated key exchange using diffie-hellman. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 156–171. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_12
11. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-46035-7_4
12. Ding, J., Alsayigh, S., Lancrenon, J., RV, S., Snook, M.: Provably secure password authenticated key exchange based on RLWE for the post-quantum world. In: Handschuh, H. (ed.) CT-RSA 2017. LNCS, vol. 10159, pp. 183–204. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52153-4_11

13. Derler, D., Slamanig, D.: Practical witness encryption for algebraic languages or how to encrypt under Groth-Sahai proofs. *Des. Codes Crypt.* **86**(11), 2525–2547 (2018)
14. Goldreich, O., Lindell, Y.: Session-key generation using human passwords only. *J. Cryptology* **19**(3), 241–340 (2006)
15. Gennaro, R., Lindell, Y.: A framework for password-based authenticated key exchange. *ACM Trans. Inf. Syst. Secur.* **9**(2), 181–234 (2006)
16. Kiltz, E., Pietrzak, K., Stam, M., Yung, M.: A new randomness extraction paradigm for hybrid encryption. In: Joux, A. (ed.) *EUROCRYPT 2009*. LNCS, vol. 5479, pp. 590–609. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_34
17. Katz, J., Ostrovsky, R., Yung, M.: Efficient and secure authenticated key exchange using weak passwords. *J. ACM* **57**(1), 1–39 (2009)
18. Katz, J., Vaikuntanathan, V.: Round-optimal password-based authenticated key exchange. *J. Cryptology* **26**(4), 714–743 (2013)
19. Shoup, V.: OAEP reconsidered. In: Kilian, J. (ed.) *CRYPTO 2001*. LNCS, vol. 2139, pp. 239–259. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_15
20. Whitfield, D., Martin, H.: New directions in cryptography. *IEEE Trans. Inf. Theory* **22**(7), 644–654 (1976)
21. Whitfield, D., Van Oorschot, P.C., Wiener, M.J.: Authentication and authenticated key exchanges. *Des. Codes Crypt.* **2**(2), 107–125 (1992)
22. Xue, H., Li, B., Lu, X.: IND-PCA secure KEM Is enough for password-based authenticated key exchange (Short Paper). In: Obana, S., Chida, K. (eds.) *IWSEC 2017*. LNCS, vol. 10418, pp. 231–241. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64200-0_14
23. Xue, H., Li, B., He, J.: New framework of password-based authenticated key exchange from only-one lossy encryption. In: Okamoto, T., Yu, Y., Au, M.H., Li, Y. (eds.) *ProvSec 2017*. LNCS, vol. 10592, pp. 188–198. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68637-0_11
24. Yao, H., Wang, C.: A novel blockchain-based authenticated key exchange protocol and its applications. In: *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 609–614. IEEE, Guangzhou (2018)
25. Zhandry, M.: How to avoid obfuscation using witness PRFs. In: Kushilevitz, E., Malkin, T. (eds.) *TCC 2016*. LNCS, vol. 9563, pp. 421–448. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49099-0_16
26. Zhu, L., Guo, C., Zhang, Z., Fu, W., Xu, R.: A Novel Contributory Cross-domain group password-based authenticated key exchange protocol with adaptive security. In: *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pp. 213–222. IEEE, Shenzhen (2017)



Certificateless Authenticated Key Agreement for Decentralized WBANs

Mwitende Gervais¹, Liang Sun², Ke Wang², and Fagen Li¹(✉)

¹ Center for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
fagenli@uestc.edu.cn

² SI-TECH Information Technology Co., Ltd., Beijing, China

Abstract. Security and privacy of sensitive data are crucial nowadays. Internet of things (IoTs) is emerging and has brought critical security issues. Wireless body networks (WBANs) as one branch of IoTs are vulnerable systems today because they carry sensitive information from implanted and wearable sensors. Authentication and key agreement for WBAN are important to protect its security and privacy. Several authentication and key agreement protocols have been proposed for WBANs. However, many of them are administered by a single server. Addition to that, a malicious key generation center can become a threat to other entities in WBANs, i.e. impersonate the user by causing a key escrow problem. In this paper, we propose a certificateless authenticated key agreement (CLAKA) for a decentralized/blockchain WBAN in the first phase. CLAKA has advantage to be designed in a decentralized architecture that is suitable for low computation devices. A security mediated signature (SMC) for blockchain authentication is described in the second phase of our protocol. SMC has advantage in solving public key revocation while maintaining the characteristics of certificateless public key cryptography i.e. solving the key escrow problem. Our protocol can compute a session key between WBAN controller and blockchain node and verify the eligibility of node to collect WBAN data.

Keywords: WBAN · Key agreement · Session key · Blockchain · SMC · SEM

1 Introduction

Wireless sensor networks are widely involved in IoT environments. WBANs play an important role in this area as well. WBAN involves directly human lives because it collects, transmits, processes and stores sensitive data. WBAN systems can collect, and process medical and non-medical data. Medical WBANs involve implanted and wearable sensors to monitor healthcare parameters such as heart diseases, temperature and blood pleasure. Non-medical WBANs that involve entertainment such as real time video streaming etc. [1]. However, the

healthcare systems face many security and privacy threats such as eavesdrop, denial of service etc. They are also made of computing capability limitations such as power, storage and memory [2]. WBAN systems need to provide data integrity and give access to the authorized nodes [3]. To achieve integrity and authentication for WBAN systems, strong authentication protocols are required; for example, certificateless authenticated key agreement. To alleviate the computation capability burden, a good choice of algorithms have to be used for example pairing free-based protocols. Apart from the cryptographic protocols, the architecture of many WBAN systems are still centralized with a weakness that they are managed by a single server.

1.1 Motivation

As discussed previously, the security and privacy are crucial for WBANs. The malicious can weaken or exploit a weak WBAN system and gain access to the sensitive information to modify it and reuse it for his own purpose. For example, an authorized access of blood glucose information can lead to the low or high level of insulin injection in the body of patient that can cause a sudden death. The data transmission and storage requires a secure environment to prevent common and uncommon known attacks, which can harm lives. Therefore, it is important in such kind of situation to design and deploy a secure authenticated key agreement among the parties before they start communicating and transmitting data. Additional to that, the service provider and storage should avoid some security risks such as single point of failure and single point of management. Since we are using body sensors on the other hand, the designed protocol requires being lightweight due to the low power computational capabilities of WBAN sensors. Most of the existing protocols present weaknesses such as key escrow problem, computation overhead as well as the system architecture that is not favorable for transmission, process, and storing of sensitive information. Hence, in this paper, a new CLAKA protocol is proposed, and a decentralized architecture is proposed to enhance the existing security protocol and architecture model for WBANs

1.2 Contribution

The major contributions of this paper are presented as follows:

- A CLAKA for blockchain-based WBAN is designed. A session key can be established between WBAN controller C and blockchain node N .
- The formal analysis of the proposed protocol is provably secure in the random oracle model (ROM) under the computation Diffie-Hellman (CDH) assumption. It can provide various security properties of CLAKA.
- A security mediated signature is used to provide authentication and verification of the data origin among the blockchain nodes. The SMC provides an instant revocation on the blockchain nodes. It is an efficient signature suitable for resource constrained devices like WBAN sensors. Compared to the

existing protocols, it provides more security attributes and can be used for WBANs.

1.3 Centralized Versus Decentralized (blockchain) WBAN

In this subsection, we briefly give an overview of the difference between a centralized and decentralized system. We first mind that most of the systems used today are centralized. Many beneficiaries do not think about the risks in revealing their identities to the social media systems. For example, when registering, a user will need to give his/her identity to the central server of a social media. Another example of a hospital that keeps data of patients and manages them as one administrator. It means that all our data and identities are managed by a central system called central-server/central authority. We trust in those systems that they will store our data privately. There is trust that the central authority will not use it for his own interests. The central system has full control over the users data. A centralized WBAN has risk of being a single point of failure without anonymity. However, the single point of management and failure can be solved by implementing a decentralized WBAN system architecture known as blockchain. Blockchain technology is a distributed, unchangeable, undeniable public ledger [4]. It means that when WBAN data is sent over a network, it is not verified by a central server. It is distributed among different nodes. The blockchain nodes are different and operate independently. An adversary will need to compromise each and every node in the blockchain, which is almost impossible. It is important to notice that nodes will not reveal their identities while doing transactions; only private keys and public key are useful i.e the anonymity is acquired in blockchain [5]. Table 1 illustrates a brief comparison between a centralized and a decentralized WBANs

Table 1. Architecture comparison

Parameter	Centralized WBAN	Decentralized WBAN
Management	WBAN data is sent to a central medical server	WBAN data is broadcast on the blockchain nodes, none has the control over the data
Trusted third part	A central server is involved between WBAN user and other users to access data	It is a distributed ledger and there is no involvement of trusted part. Transactions are broadcast on the network, validated by nodes and are added or denied
Single point of failure	One central server takes control and it is easy to failure during the operation time	Transaction are broadcast on different nodes. Every node keeps the same block. If one node failure others will continue operating
Anonymity	Anonymity is not maintained	Anonymity is maintained

1.4 Organization

The remaining part of the paper is organized as the following. In Sect. 2, the related works are presented. In Sect. 3, the preliminaries are discussed. In Sect. 4, the modeling of CLAKA scheme is presented. In Sect. 5, the proposed protocol is designed. Section 6, the analysis of the proposed protocol is discussed. In Sect. 7, we conclude our work.

2 Related Works

Authentication and key agreement for WBANs have been of great concern recently. Most of cryptographic protocols have some weakness such as in system architectures, computation overhead, data processing, and storage management problem. For this reason, we exploit some existing authenticated and key agreement protocols related to medical/or wireless body area networks. Researcher have put effort in designing suitable protocols for WBANs, example Shen et al. [6] designed a lightweight multi-layer authentication protocol for WBANs. Their protocol is a certificateless protocol without pairing based on elliptic curve cryptography. The first layer includes communication between body sensors and personal digital assistant (PDA), and the second layer takes communication part between PDA and application provider. Their protocol achieves different security features based on intractability of elliptic curve discrete logarithm problem (ECDHP), CDH, and elliptic curve factorization problem (ECFP) assumptions. It computes many points multiplication, and WBAN data is managed by a central server. Again Shen et al. [7] proposed a cloud-aided lightweight certificateless authentication protocol with anonymity for WBANs. Their protocol consists of a three tier architecture. Tier 1 is for intra-BAN between body sensor and PDA, tier 2 is the communication between PDA and access point and tier 3 provides communication between cloud and WBAN network. The protocol can establish a session key. It also achieves many security features like anonymity but it is computationally cost. The cost for the proposed equipment can be high due to the combination of management server and cloud servers at the same time. The number of hash functions used needs to be reduced and it is a centralized architecture. Li et al. [8] designed an enhanced 1-round authentication protocol for WBANs. It provides mutual authentication and session key. The protocol solves key escrow problem, resistant to offline-password guessing attack and it is proved to be secure in BAN logic. Li et al. [9] proposed an efficient anonymous authenticated key agreement protocol for WBANs. Their protocol achieves different security features like non-repudiation and adds the key update property, but it is a pairing-based protocol. Wazid et al. [10] proposed an authenticated key management protocol for cloud-assisted body area sensor networks. Their protocol can achieve different security features such ephemeral secret leakage resiliency. It is proved to be secure in real-or-random (ROR) model. In 2018, Wazid et al. [11] proposed a novel authentication and key agreement protocol for implantable medical devices deployment. Their protocol achieves many security features such as anonymity and untraceability. It is proved to be secure with

automated validation of internet security protocols and applications (AVISPA) tool.

3 Preliminaries

3.1 Notations

Notations and their meaning used for CLAKA protocol are described in the following table (Table 2).

Table 2. Notations and description

Notation	Description
P_0	The public key of KGC
s	Master secret key
i	i^{th} Users
Q_i	The partial private key of user i
ID_i	The user's identity
u_i	The secret value of user i
X_i	The public key of user i
d_i	The private key of user i
K_{ij}, K_{ji}	The session key of user i and user j
$a, b, y,$ and r_i	random numbers
p	A large prime number
\mathbb{F}_p	Prime field
E	An elliptic curve E over a prime field \mathbb{F}_p
G	Additive group
P	Generator of G
f_1, f_2, f_3	Hash functions
SEM	Security mediator
p, q	Prime numbers
x_N	The node's private key
P_N	The node's public key
D_S	The SEM private key

3.2 Elliptic Curve

Let E/\mathbb{F}_p be elliptic curve E over a finite field \mathbb{F}_p , defined by the equation:

$$y^2 = (x^3 + ax + b), a, b \in \mathbb{F}_p \quad (1)$$

the discriminant

$$\Delta = (4a^3 + 27b^2) \neq 0. \tag{2}$$

The points on E/\mathbb{F}_p , and the point at infinity make a group of points G

$$G = \{(x, y) : x, y \in \mathbb{F}_p, E(x, y) = 0\} \cup \{O\}. \tag{3}$$

Assume q to be the order of G . The scalar multiplication over E/\mathbb{F}_p is defined as

$$tP = P + P + P + \dots + P \text{ (} t \text{ times)}. \tag{4}$$

The detailed mathematical operations related to elliptic curve can be found in [12]. The following defined problems over G are assumed to be intractable within polynomial time.

3.3 Hard Assumptions

Definition 1. *DLP assumption: Given (P, aP) , for an unknown selected value $a \in \mathbb{Z}_q^*$ and P generator of \mathbb{G} , compute aP . The DLP states that it is intractable to determine the value a for any probabilistic polynomial-time.*

Definition 2. *CDH assumption: Given (P, aP, bP) , for unknown $a, b \in \mathbb{Z}_q^*$ and P generator of \mathbb{G} , compute abP . The CDH hard assumption states that for any probabilistic polynomial-time, it is intractable to solve the CDH problem.*

3.4 Algorithms for CLAKA Protocol

A CLAKA protocol consists of the following six algorithms.

- **Setup:** A KGC takes j as input security parameter, output a master key and system parameter list $pars$.
- **Partial-Private-Key-Extract:** A KGC takes as input $pars$, a master key, and a user identity ID_i to return a user partial private key Q_i .
- **Set-Secret-Value:** It takes as input $pars$ and a user ID_i to return user’s secret value u_i .
- **Set-Private-Key:** It takes as inputs $pars, ID_i$, a partial private key Q_i , and a secret value u_i to return a private key d_i for the user.
- **Set-Public-Key:** It takes as inputs $pars$, user ID_i , and the secret value u_i to return a public key X_i for the user.
- **Key-Agreement:** It is a polynomial participative algorithm for both users/nodes. It takes as inputs $pars$ for controller C and node N , with (d_C, ID_C, X_C) for controller C , and (d_N, ID_N, X_N) for node N ; where d_C and d_N are private keys for controller C and node N ; ID_C and ID_N are identities for controller C and node N . The X_C and X_N are set to be public key of controller C and node N . Finally both users compute a session key $K_{CN} = K_{NC} = K$.

3.5 Security Mediated Signature

Traditionally, the architecture of digital signature is linked to relatively high risk for the signer, for the recipient and for the service provider. Such risks have direct influence on the costs for all participants and on very low popularity of qualified signatures in business. Infact, the only usage of signatures is due to legal obligations in contact with public bodies. The risks can be easily solved, if we organize the system architecture. A good solution is using a mediated signature [13]. The mechanism for mediated involves multiple keys, atleast two keys to create a signature. For a public key K there is a pair k_1 and k_2 . The security of a mediated signature is described in a way that having K , k_1 any number of signature created with k_1 and k_2 , it is infeasible to derive k_2 . Having K , k_2 and any number of signatures created with k_1 and k_2 , and any number of presignatures created with k_1 is infeasible to derive k_1 . Without k_1 or k_2 , it is infeasible to create a valid signature. Security mediated signatures have many applications for example signatures for public administration to solve authorization problem of data submitted by citizens to the public agents. In this scenario, a server called mediator that holds the second key for each user. Another example in wireless sensor networks which are deployed for variety of applications in smart cities, monitoring of air pollution, traffic jam, power, and transportation systems [13]. Due to its lightweight in computation for our case, a security mediated signature can be applied and is suitable for WBAN applications as shown bellow in Sect. 5.

Definition 3. *A security mediated signature maintains the security features of certificateless signature and adds key revocation capability.*

We adopt a security mediated signature [14] for node authentication on decentralized WBAN. The eligible node N can be authenticated and verified before the ledger is added to the chain. The signature consists of the following five algorithms.

- **Setup:** This algorithm take as input a security parameter j and returns a master key s and system parameter list $pars$.
- **Key-Gen:** This algorithm takes as inputs $pars$. Selects a random value, and computes the private/public key pair (x_i, P_i) .
- **Register:** This algorithm takes as inputs $pars$, master secret s , user's ID_i and public key P_i to outputs a SEM private signing key d_i .
- **Sign:** It is a participative probabilistic algorithm between user and SEM. They have in common $pars$, a message μ , user ID_i . The SEM also adds d_i to run the part 1 algorithm SEM-Sign. User adds in particular x_i to run the part 2 algorithm User-sign. Both parties ends up with computing a signature φ or \perp when SEM does not give a valid signature, for example when the user signing capability has been revoked.
- **Verify:** This algorithm takes $pars$, a message μ , a user ID_i , user public key P_i and signature φ ; it outputs true or false. The concrete security mediated signature is presented in Sect. 6 for node authentication and verification.

4 Modeling CLAKA Protocol

Before describing the CLAKA protocol model, we first review the two types of adversaries in certificateless cryptography. The CLAKA protocol requires to be resistant to two types of attacks said Type I and Type II adversaries as described in [16].

- **Type I Adversary A_1 :** The A_1 does not have access to the master secret key, but can replace public key of any party with a value of his choice.
- **Type II Adversary A_2 :** The A_2 has access to the master secret key but can not replace public key of any party.

4.1 Informal Security Definitions

We define informally the security requirement for CLAKA protocols. The key secrecy resiliency is the core security property for CLAKA. This means that adversary has no ability to learn a session key negotiated between participants [15]. Addition to that the following properties are useful in CLAKA protocols based on decentralized (blockchain) system. There are partially similar to the He et al. [16].

- **Unknown key share:** The session key is created at every session. The awareness of the previously created key cannot allow adversary to reveal the next session keys.
- **Key compromise impersonation:** A compromised private key of an entity C does not allow the attacker to impersonate other entity N to C .
- **Key control:** No one among the entities is able to force a preselected session key.
- **Key escrow:** The KGC cannot know the user's private key. Users are the only entities to know their secret values; with these secrets any adversary cannot compute their private keys.
- **Anonymity:** This requirement ensures that an adversary does not get the identities of legal users in authentication process. Data detection, collection and transmission of data is closely related with the user in WBAN. These data refers to the user's private information, so users want to use their own medical services, and at the same time their privacy will not be disclosed to the unauthorized illegal third party. Therefore, the purpose of anonymity is to protect the user from being compromised when using the service.
- **No-repudiation:** This requirement performs that the users cannot deny their own use of the service, while service providers cannot deny that they provide the certain service to the user. The blockchain nodes computes the signature information with SEM to be verified about its authenticity.
- **Immutability:** It is an ability where a blockchain ledger remains permanent. No one can modify a decentralized ledger of the committed blocks.
- **Revocation:** Once the identity or key of user is thought to have been compromised the security mediator can not sign the message.

- **Verifiability.** Blockchain transactions are publicly known to nodes everywhere. Anyone can check the transactions and hash functions along way back to the previous block.
- **Consensus mechanism.** It is a mechanism used in blockchain to make an agreement on data value or network state among the distributed processes. It is important in keeping records.

4.2 Formal Security Model

Encouraged by Zhang et al. [15], we describe the formal security model for CLAKA. It is modeled as the game between challenger \mathcal{C} and adversary $A \in \{A_1, A_2\}$. The adversary monitors all interactions between two parties. Every party possesses an identity ID_i . The characteristics of A are represented by the number of oracles kept by \mathcal{C} . Assume that an oracle $\phi_{i,j}^r$ represents r^{th} instance of party i and his counterpart j in a session. The game starts when \mathcal{C} sets up algorithm with security parameter λ to return master secret and system $params$. If A is Type I adversary A_1 , \mathcal{C} transmits $params$ to A and maintains master key secret; else A is Type II adversary A_2 , \mathcal{C} issues $params$ and master key to A . Adversary A is a probabilistic polynomial time Turing machine. All communications go through A . Parties answer to the queries from A and do not interact between them. A acts as benign, i.e. A is deterministic and prefers to choosing two oracles $\phi_{i,j}^n$ and $\phi_{j,i}^l$ and takes each message from one oracle to another. In addition, A can ask for the following queries, including one **Test** query in the following way:

- **Create**(ID_i): This query permits A to request \mathcal{C} to create a new party i whose identity is ID_i . Upon receiving such query, \mathcal{C} creates private and public keys for i .
- **Public-Key**(ID_i): A can may ask for the public key of a party i whose identity is ID_i . To answer, \mathcal{C} replays with the public key X_i of party i .
- **Partial-Private-Key**(ID_i): A may ask for partial private key of party i whose identity is ID_i . To answer, \mathcal{C} replays with partial private key Q_i of party i .
- **Corrupt**(ID_i): A may ask for the private key of party i whose identity is ID_i . To answer, \mathcal{C} replays with the private key d_i of party i .
- **Public-Key-Replacement**(ID_i, X'_i): For a party i whose identity is ID_i ; A may select another public key X' and set X' as the public key. \mathcal{C} documents this change to be used in the future.
- **Send**($\phi_{i,j}^n, \mu$): A may select and issues a message μ to an oracle $\Phi_{i,j}^n$, by which, a party i supposes to be sent from party j . A can also create a particular **Send** query with $\mu \neq \alpha$ to an oracle $\Phi_{i,j}^n$, which tells i to start a protocol runs with j . It is called an initiator oracle when the first message it has obtained is α . Otherwise, it is called a responder oracle.
- **Reveal**($\phi_{i,j}^n$): A may request a special oracle to reveal the session key, if any, it currently holds to A .
- **Test**($\phi_{i,j}^n$): At certain level, A can choose one of the oracles, for example $\Phi_{1,j}^T$ to request for one **Test** query. Such oracle should be *fresh*. To answer the

query, the oracle guesses a coin $b \in \{0, 1\}$, and outputs the session key held by $\Phi_{I,J}^T$ if $b = 0$, or a random sample from the distribution of session key if $b = 1$.

An oracle $(\phi_{i,j}^n)$ can be set to one of the three states

- *Accepted*: An oracle is in *Accepted* state if it has accepted the request to create a session key.
- *Rejected*: An oracle is in *Rejected* state if it has rejected the request to create a session key.
- *State**: If none of the previous states decision has been taken.
- *Opened*: If an oracle has answered the **Reveal** query.

Definition 4. *A matching conversation: Two oracles $(\phi_{i,j}^n)$ and $(\phi_{j,i}^l)$ have a matching conversation if they have identical session key.*

Definition 5. *Fresh Oracle: An oracle $(\phi_{i,j}^n)$ is fresh if it is in the accepted state; or it is not in the opened state; or party $j \neq i$ is not corrupted; or $(\phi_{j,i}^l)$ does not exist in opened state to have the matching conversation with $(\phi_{i,j}^n)$; or if A is Type I and has not requested the private key of party j and if A is Type II and has not replaced the public key of party j .*

The fresh oracle definition can allow party i to be corrupted so it is used to solve the key compromise impersonation attack.

After a **Test** query, A may go on to query the oracles except make **Reveal** query to test oracle $\Phi_{I,J}^T$, or to $\Phi_{J,I}^l$ who has a matched conversation with $\Phi_{I,J}^T$, and it can not corrupt the user J . In addition, if A is Type I, A can not ask for partial private key of the participant J ; and if A is a Type II adversary, J cannot replace the public key of the user J . At the end of the game, A must output a guess bit b' . A wins if and only if $b' = b$. A 's advantage to win the game, is defined as:

$$A^j = \left| \Pr[b' = b] - \frac{1}{2} \right| \quad (5)$$

Definition 6. *A CLAKA protocol is secured if:*

- *In the presence of a benign adversary on $\Phi_{i,j}^n$ and $\Phi_{j,i}^l$, both oracles always agree on the same session key, and this key is distributed uniformly at random.*
- *For an adversary A , advantage A^j of winning game is negligible.*

5 The Proposed Protocol

The proposed protocol consists of two main parts. The first part consists of a new certificateless authenticated key agreement for WBAN; another part consists of node authentication and verification where we can apply an adopted security mediated signature. Both parts are presented into the following Subjects. 5.2 and 5.3.

5.1 System Model

The proposed model for blockchain-based WBAN, includes three entities (KGC, Controller C and Blockchain node N). The KGC registers and computes partial private keys for controller and node N . Upon receiving partial private keys, both users establish a session key to authenticate users and secure data transmission. Note that the key is established every session when the two entities want to communicate to avoid the known key share problem. Figure 1 illustrates the proposed system model architecture as explained in the following steps.

- The KGC is dedicated to register the controller C and decentralized node N . Also, it generates system parameter list. KGC cannot know about the private keys of C and node N .
- The Controller collects WBAN data from different body sensors, and transmit them to the blockchain node via wireless network. The data is supposed to be protected. Before sending data to the blockchain, C computes its private key and establishes a session key with blockchain node N . The session key will encrypt data transmitted from C to blockchain node N .
- The blockchain node N works as the collector node. It is incharge of collecting data coming from controller and broadcast it to the blockchain nodes. Blockchain node N should be registered with KGC and get partial-private key and system parameters. It also establishes a session key with C . The session key is used to encrypt and decrypt data that is sent to the blockchain.

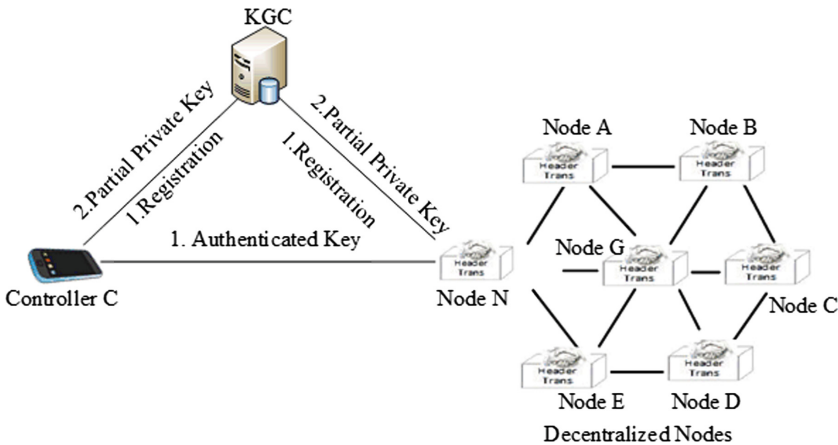


Fig. 1. System model

5.2 The Proposed CLAKA for Decentralized WBAN

In this section, a CLAKA scheme is proposed. It consists of six polynomial time algorithms. They are presented as follows.

- **Setup:** This algorithm takes security parameter j as its input and returns system parameters and master key. KGC performs the following operations.
 1. Given a security parameter j , KGC selects an additive group G of prime order q and P is a generator of the group.
 2. Selects a random master key $s \in \mathbb{Z}_q^*$ and calculates $P_0 = sP$ as master public key.
 3. Selects hash functions $f_1 : \{0, 1\}^* \times G \rightarrow \mathbb{Z}_q^*$ and $f_2 : \{0, 1\}^* \times \{0, 1\}^* \times G \times G \times G \times G \rightarrow \{0, 1\}^j$.
 4. KGC publishes system params $(\mathbb{F}_p, E/\mathbb{F}_p, G, q, P, P_0, f_1, f_2)$ and keeps s secret.
- **Partial-Private-Key-Extract:** It takes as inputs $pars$, the master key s and user identity ID_i and returns partial private key of users as follows
 1. KGC selects a random number $e_i \in \mathbb{Z}_q^*$ computes $R_i = e_iP$, $h_i = f_1(ID_i, R_i)$.
 2. KGC computes $s_i = (e_i + sh_i) \bmod q$.
 3. KGC sets $Q_i = (s_i, R_i)$ as user's partial private key.
 4. User i verifies whether the partial private key is valid by computing the equation $s_iP = R_i + f_1(ID_i, R_i)P_0$.
- **Set-Secret-Value:** This algorithm takes $pars$ and user's ID , selects randomly $u_i \in \mathbb{Z}_q^*$. u_i is sets as secret value.
- **Set-Private-Key:** The algorithm takes as inputs $pars$, partial private key Q_i , user's ID_i , and secret value u_i and returns user's private key $d_i = (u_i, Q_i)$.
- **Set-Public-Key:** The algorithm takes as input $pars$, user ID_i and user's secret value u_i to return user's public key $X_i = u_iP$.
- **Key-Agreement:** Assuming that any node of blockchain can establish an authenticated key agreement with WBAN controller C . Lets node N and C establish an authenticated key, and one is initiator another one responder. The controller C with identity ID_C possesses the private key $d_C = (u_C, Q_C)$ and the public key $X_C = u_CP$. The blockchain node N with identity ID_N possesses the private key $d_N = (u_N, Q_N)$ and the public key $X_N = u_NP$. The controller C and node N the protocol computes as follows:
 1. Controller C selects $a \in \mathbb{Z}_q^*$, computes $T_C = aP$ and sends a message (ID_C, T_C) to the N .
 2. N selects $b \in \mathbb{Z}_q^*$, computes $T_N = bP$ and sends a message (ID_N, T_N) to C .

Both C and N can compute the secrets as the following:

C computes

$$K_C^1 = a(R_N + h_N P_0) + s_C T_N + a X_N + u_C T_N = K_1 \text{ and } K_C^2 = a.T_N = K_2 \quad (6)$$

N computes

$$K_N^1 = b(R_C + h_C P_0) + s_N T_C + b X_C + u_N T_C = K_1 \text{ and } K_N^2 = b.T_C = K_2 \quad (7)$$

Correctness

$$\begin{aligned}
K_C^1 &= a(R_N + h_N P_0) + s_C T_N + bX_N + u_C T_N \\
&= a(e_N P + sh_N P) + bs_C P + au_N P + bu_C P \\
&= a(e_N + sh_N)P + bs_C P + u_N T_N + bX_C \\
&= s_N T_C + b(e_C + sh_C)P + bX_C + u_N T_C \\
&= s_N T_C + b(R_C + h_C P_0) + bX_C + u_N T_C \\
&= s_N T_C + b(e_C + sh_C)P + bX_C + x_N T_C \\
&= s_N T_C + b(R_C + h_C P_0) + bX_C + u_N T_C \\
&= b(R_C + h_C P_0) + s_N T_C + bX_C + u_N T_C \\
&= K_N^1 \\
&= K_1 \\
K_C^2 &= aT_N \\
&= abP \\
&= baP \\
&= K_N^2 \\
&= K_2
\end{aligned}$$

The established session key $K = f_2(ID_C, ID_N, T_C, T_N, K_1, K_2)$.

5.3 Node Authentication and Verification

In this section, we describe an adopted security mediated signature from [14]. It is a signature without pairing based on intractability of discrete logarithm problem that has advantage of authentication, verifiability, and instant revocation. The security mediated signature scheme consists of the following algorithms.

- **Setup.** Given the security parameter j , KGC performs the following operations:
 1. Generate two primes p and q such that $q|p-1$.
 2. Selects P as generator of set \mathbb{Z}_q^* .
 3. Selects randomly $s \in \mathbb{Z}_q^*$ and computes $Y = sP$.
 4. Chooses hash functions $f_1 : \{0, 1\}^* \rightarrow \{0, 1\}^j$, $f_2 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$ and $f_3 : \{0, 1\}^* \rightarrow \mathbb{Z}_q^*$.
 5. Outputs $params = \{p, q, P, Y, f_1, f_2, f_3\}$ and master key s is kept secret.
- **Key-Gen.** The user/node performs the following operations.
 1. Selects $x_N \in \mathbb{Z}_q^*$ as the user private key.
 2. Computes $P_N = x_N P$ as node's public key.
- **Register.** The user/node performs the following operations.
 1. The KGC authenticates and register node identification(ID_N, P_N) key and chooses randomly $w \in \mathbb{Z}_q^*$.
 2. Computes $W = wP$ and $d_S = w + sf_2(ID_N, W) \bmod q$.

3. The KGC sends SEM private key $D_S = (W, d_S)$ and (ID_N, P_N) to the SEM over an authentic and secure channel.
- **Sign.** The node N can request for a signature over a message μ , SEM checks whether ID_N is not revoked; the communication between node and SEM starts as follows:
 1. SEM-Sign (1). Randomly chooses $r_S \in \mathbb{Z}_q^*$, calculates $R_S = r_S P$, and sends $c = f_1(R_S)$ to the node N .
 2. Node-Sign (1): Randomly chooses $r_N \in \mathbb{Z}_q^*$, sends $R_N = r_N P$ to the SEM.
 3. SEM-Sign (2). computes $R = R_S + R_N$, $h_S = f_3(ID_N, W, 0, P_N, R, \mu)$, and $t = r_S + d_S h_S \bmod q$ and returns back (R_S, t) to the node. Note that the partial signature generated by the SEM will not require to be sent over a secure channel because none can get advantage on it. The partial signature given by the SEM can be verified by computing $R_S = tP + (-h_S)(W + Y f_1(ID_N, W))$ holds.
 4. Node-Sign (2). First checks $c = f_1(R_S)$, if they are equal, computes $R = R_S + R_N$, $h_N = f_3(ID_N, P_N, 1, W, R, \mu)$, and $v = (r_N + x_N \cdot h_N + t) \bmod q$.
The complete signature $\varphi = \langle R, v, (ID_N, X_N, W) \rangle$.
 - **Verify.** Given φ , accepts if and only if the following equality holds.

$$R = vP + (-P_N h_N) + (-h_S(Y + W f_1(ID_N, W))). \quad (8)$$

Correctness

The correctness of security mediated signature is verified as follows.

$$\begin{aligned}
 R^* &= vP + (-h_N P_N) + (-h_S)(W + Y f_2(ID_N, W)) \\
 &= vP + (-h_N(x_N P)) + ((-h_S)wP + (-h_S)sP f_2(ID_N, W)) \\
 &= (r_N + x_N h_N + t)P + (h_N x_N P) + ((-h_S)wP) + (-h_S)sP f_2(ID_N, W) \\
 &= (r_N P + x_N h_N P + tP) + (-h_N x_N P) + (-h_S wP + h_S sP f_2(ID_N, W)) \\
 &= r_N P + x_N h_N P + tP - h_N x_N P + (-w h_S P - h_S sP f_2(ID_N, W)) \\
 &= r_N P + tP + (-h_S P(w + s f_2(ID_N, W))) \\
 &= r_N P + tP + (-h_S d_S P) \\
 &= r_N P + (r_S + h_S d_S)P + (-h_S d_S P) \\
 &= r_N P + r_S P \\
 &= R_N + R_S \\
 &= R
 \end{aligned}$$

Data Encryption

- **Consensus message.** Before sending data to the blockchain, nodes will need a permission from controller to use WBAN data. The controller first needs to send a consent message to the blockchain nodes that allows them to use WBAN data, otherwise they can not use data without the owner's permission.

The node N receives a consensus message from C and broadcasts it to the blockchain. Controller sends a message μ to the blockchain by encrypting it using a session key K as follows:

$$(K||\mu)$$

N obtains the encrypted message μ and uses the session key K to recover the consent form to use WBAN data from C . N deletes K .

- **Message broadcast.** Node N signs and broadcasts a consensus message to the blockchain using its security mediated signature. Any node can verify the authenticity of node N and add a consent message to the ledger.

6 Security Analysis

This section includes the formal security analysis, informal discussion of security properties and comparison with existing protocol.

6.1 Formal Analysis

The security analysis of the proposed protocol relays on CDH assumption. We follow the security prove described in [17]. The CDH hard assumption in group G is given. Two random oracles f_1 and f_2 use an idea explained in [18]. For security prove, we follow theorems and lemma given bellow.

Theorem 1. *The proposed protocol is a secure CLAKA protocol.*

Proof: The CLAKA protocol is proved to be secure against two types of adversaries. The proof of Theorem 1 is discussed in the following Lemmas 1, 2 and 3.

Lemma 1. *In the existence of benign adversary, two matching oracles $\phi_{i,j}^n$ and $\phi_{j,i}^k$ establish the identical session key as if there is no adversary. The session key is distributed evenly at random.*

Proof. Suppose that i and j are two users in the protocol and A is a benign adversary. In this situation, the two oracles gets correctly identical message to the original messages from other oracle; therefore, they consent on the same session key. Since a and b are chosen at random by user i and j , the session key can be taken as the output of hash function f_2 on a random input. Based on the properties of hash function, the session key is uniformly distributed over $\{0, 1\}^J$. As it is detailed in our protocol correctness. The numbers a and b are randomly chosen, two oracles are matching, they are authorized either and the session key is consistently shared.

Thus controller C computes

$$f_2(ID_C, ID_N, T_C, T_N, K_C^1, K_C^2)$$

And node N computes

$$\begin{aligned}
 &f_2(ID_C, ID_N, T_C, T_N, K_N^1, K_N^2) \\
 &K_C^1 = K_N^1 = K_1 \\
 &K_C^2 = K_N^2 = K_2
 \end{aligned}$$

Finally the matching oracles compute the session key

$$K = f_2(ID_C, ID_N, T_C, T_N, K_1, K_2)$$

Lemma 2. *Suppose that CDH problem is intractable, the advantage of a Type I adversary in winning game is negligible in the ROM.*

Proof. Assume that A can make at most q_{f_2} times f_2 queries and create at most q_c parties. Advantage for A to win the game is A^j . Therefore, the challenger can solve the CDH problem with the advantage $\frac{1}{q_c^2 q_s q_{f_2}} A^j$, q_s is the number of sessions every party can be involved in at most.

Assuming that a Type I adversary A can win with a non negligible advantage A^j in polynomial time t . We demonstrate that challenger \mathcal{C} can solve CDH problem with a non negligible probability. We demonstrate how challenger \mathcal{C} use A to compute abP .

All adversary's queries now pass through \mathcal{C} . The game is initiated when \mathcal{C} selects a and sets $P_0 = aP$; \mathcal{C} selects at random $I, J \in [1, q_{f_1}], T \in [1, q_s], s_I, u_I, h_I \in \mathbb{Z}_q^*$ and computes $R_I = s_I P, X_I = u_I P$, and sets P_0 as the system public key and sends system $params = \{G, P, P_0, f_1, f_2, j\}$ to A .

- **Create(ID_i):** A challenger \mathcal{C} maintains an empty list L_c initially consisting of the tuples (ID_i, Q_i, u_i, X_i) . If $ID_i = ID_I$, challenger \mathcal{C} let's partial private key, private key and public key to be $Q_i = (s_I, R_I)$, $d_i = (u_I, Q_I)$ and X_I separately. Challenger \mathcal{C} also lets $f_1(ID_I, R_I) \leftarrow h_I$ where R_I, u_I, h_I are mentioned above. Otherwise, challenger \mathcal{C} chooses randomly $u_i, s_i, h_i \in \mathbb{Z}_q^*$ and computes $R_i = s_i P - h_i P_0$, public key is $X_i = u_i P$, then i 's partial private key $Q_i = (s_i, R_i)$, private key $d_i = (u_i, Q_i)$ and public key X_i . Finally adds the tuples (ID_i, Q_i, u_i, X_i) and (ID_i, R_i, X_i, h_i) to the list L_c and L_{f_1} separately.
- f_1 query: Challenger \mathcal{C} keeps initial empty list L_{f_1} which has tuples of the form (ID_i, R_i, X_i, h_i) . If (ID_i, R_i, X_i) is on the list L_{f_1} , then h_i is returned. Else, challenger \mathcal{C} executes the query **Create(ID_i)** and returns h_i .
- **Public-Key(ID_i):** Upon obtaining such query, challenger \mathcal{C} looks for a tuple (ID_i, Q_i, u_i, X_i) in the list L_c indexed by ID_i , and outputs X_i as response.
- **Partial-Private-Key(ID_i):** Once a challenger \mathcal{C} is given such query, if $ID_i = ID_I$, \mathcal{C} aborts. Otherwise, \mathcal{C} looks for a tuple (ID_i, Q_i, u_i, X_i) in a list L_c indexed by ID_i , and outputs Q_i as response.
- **Corrupt(ID_i):** Once a challenger \mathcal{C} is given such query, if $ID_i = ID_I$, \mathcal{C} aborts; else, \mathcal{C} looks for a tuple (ID_i, Q_i, u_i, X_i) in a list L_c indexed by ID_i , if $u_i = \perp$, challenger \mathcal{C} outputs \perp . Else challenger \mathcal{C} gives (u_i, Q_i) as response.

- **Public-Key-Replacement**(ID_i, X_i'): If $ID_i = ID_I$, \mathcal{C} aborts. Otherwise, challenger \mathcal{C} looks for a tuple (ID_i, Q_i, u_i, X_i) in L_c indexed by ID_i and upgrades X_i to X_i' and sets $u_i = \perp$.
- **Send**($\Phi_{i,j}^n, \mu$): Challenger \mathcal{C} keeps empty list L_s consisting of tuples of the form $(\Phi_{i,j}^n, r_{i,j}^n, \mu_{i,j}^n, \mu_{j,i}^n, X_i^n, X_j^n, SK_{i,j}^n)$, where $\mu_{j,i}^n$ is the coming message, X_j^n is the public key of the participant j received by $\Phi_{i,j}^n$, X_i^n is the current public key owned by the user i , $r_{i,j}^n, \mu_{i,j}^n$ are described below. Upon receiving such query, if $\mu \neq \alpha$, challenger \mathcal{C} sets $\mu_{j,i}^n = \mu$; else at the end of protocol, a message will be returned. If $\Phi_{i,j}^n$ is accepted, challenger sets message to be $\mu_{j,i}^n$ and similar response from L_s is given once the query has been requested before, if not the challenger does as the following:
 1. If $n = T$, $ID_i = ID_I$, $ID_j = ID_J$, challenger \mathcal{C} sets $SK_{i,j}^n = r_{i,j}^n = \perp$ sets $\mu_{i,j}^n = \alpha P$, return $\mu_{i,j}^n$ as the answer and adds the tuple $(\Phi_{i,j}^n, r_{i,j}^n, \mu_{i,j}^n, \mu_{j,i}^n, X_i^n, X_j^n, SK_{i,j}^n)$ to the list L_s .
 2. Else, if $ID_i \neq ID_J$, selects a random $r_{i,j}^n \in Z_n^*$, computes $\mu_{i,j}^n = r_{i,j}^n P_0$, returns $\mu_{i,j}^n$ as the response, sets $SK_{i,j}^n = \perp$ and adds $(\Phi_{i,j}^n, r_{i,j}^n, \mu_{i,j}^n, \mu_{j,i}^n, X_i^n, X_j^n, SK_{i,j}^n)$ to the list L_s .
 3. Else, selects a random $r_{i,j}^n \in Z_n^*$, computes $\mu_{i,j}^n = r_{i,j}^n P$, returns $\mu_{i,j}^n$ as the response, sets $SK_{i,j}^n = \perp$, and adds $(\Phi_{i,j}^n, r_{i,j}^n, \mu_{i,j}^n, \mu_{j,i}^n, X_i^n, X_j^n, SK_{i,j}^n)$ to the list L_s .
- **Reveal**($\Phi_{i,j}^n$): Once receive such query, challenger \mathcal{C} calls L_s for a tuple $(\Phi_{i,j}^n, r_{i,j}^n, \mu_{i,j}^n, \mu_{j,i}^n, X_i^n, X_j^n, SK_{i,j}^n)$, sets $\mu_{i,j}^n = T_i$ and $\mu_{j,i}^n = T_j$ if $SK_{i,j}^n \neq \perp$, then challenger \mathcal{C} returns $SK_{i,j}^n$ as the response. Otherwise, challenger \mathcal{C} looks for the tuple (ID_i, Q_i, u_i, X_i) on the list L_c and does the following:
 - If $n = T$, $ID_i = ID_I$, $ID_j = ID_J$ or $(\Phi_{i,j}^n)$ is oracle which has the matched conversation with $(\Phi_{I,J}^T)$, challenger \mathcal{C} aborts.
 - Else if $ID_i \neq ID_I$, there are two steps:
 1. Challenger \mathcal{C} looks in the list L_{f_2} and L_c for the corresponding tuples $(ID_i, ID_j, T_i, T_j, X_i, X_j K_{i,j}^1, K_{i,j}^2, h_u)$ and (ID_i, Q_i, u_i, X_i) , then computes $K_{i,j}^1 = r_{i,j}^n (R_i + h_i P_{pub}) + s_i T_{j,i}^n$, $K_{i,j}^2 = r_{i,j}^n T_{j,i}^n$.
 2. Otherwise, randomly sample $SK_i \in \{0, 1\}^J$ and return $SK_{i,j}^n$ as the answer.
- *f₂ query*: Challenger \mathcal{C} maintains a list L_{f_2} of the form $(ID_u^i, ID_u^j, T_u^i, T_u^j, K_u^1, K_u^2, h_u)$ and responds with f_2 queries $(ID_u^i, ID_u^j, T_u^i, T_u^j, K_u^1, K_u^2)$ in the following ways:
 1. If a tuple indexed by $(ID_u^i, ID_u^j, T_u^i, T_u^j, K_u^i, K_u^j)$ is already in L_{f_2} , challenger responds with the corresponding h_u .
 2. Else challenger \mathcal{C} chooses $h_u \in \{0, 1\}^J$. Challenger \mathcal{C} chooses $h_u \in \{0, 1\}^J$ and add the tuple $(ID_u^i, ID_u^j, T_u^i, T_u^j, K_u^i, K_u^j, h_u)$ to the list L_{f_2} .
- **Test**($\Phi_{i,j}^n$): At some level, challenger \mathcal{C} will ask a test query on some oracles. If challenger \mathcal{C} does not select one of the oracles $\Phi_{I,J}^T$ to ask the **Test** query,

then \mathcal{C} aborts. Otherwise, \mathcal{C} only outputs a random value $b \in \{0, 1\}^J$. The probability that \mathcal{C} selects $\Phi_{I,J}^T$ as the **Test** oracle is $\frac{1}{q_c^2 q_s}$. For this case, challenger \mathcal{C} wouldn't have made $\text{Corrupt}(\Phi_{I,J}^T)$ or $\text{Reveal}(\Phi_{I,J}^T)$ queries, and so challenger \mathcal{C} would not have aborted. If challenger \mathcal{C} can win in such a game, then challenger \mathcal{C} must have made the corresponding f_2 query of the form $(ID_T^i, ID_T^j, T_T^i, T_T^j, K_T^1, K_T^2)$. If $\Phi_{I,J}^T$ is the initiator oracle or else $(ID_T^i, ID_T^j, T_T^i, T_T^j, K_T^1, K_T^2)$, with overwhelming probability because f_2 is a random oracle. Thus \mathcal{C} can find the corresponding item in the f_2 list with probability and $\frac{1}{q_{f_2}}$ and outputs $K_T^1 - s_I aP - r_{I,J}^T (R_J + h_J P_{pub})$ as a solution to the CDH problem. The probability that \mathcal{C} solves the CDH problem is $\frac{\varepsilon}{q_c^2 q_s q_{f_2}}$.

Lemma 3. *Under the assumption that the CDH problem is intractable, the advantage of a Type II adversary A_2 against our protocol is negligible in the random oracle model.*

Proof. Suppose that there is a Type II adversary A_2 who can win the game defined in Sect. 4, with a non-negligible advantage A^j in polynomial time t . Then, A_2 can win the game with no-negligible probability ε . We show how to use the ability of A_2 to construct an algorithm \mathcal{C} to solve the CDH problem. Suppose a challenger \mathcal{C} is given an instance (aP, bP) of the CDH problem, and wants to compute cP with $c = ab \bmod q$. \mathcal{C} first chooses $s \in G$ at random, sets sP as the system public key P_0 , selects the system parameter params $\langle \mathbb{F}_p, E/\mathbb{F}_p, G, P, P_0, f_1, f_2 \rangle$, sends params and master key s to A_2 . Supposed A_2 makes at most q_{f_i} times f_i queries and creates at most q_c participants. Let q_s be the maximum number of sessions each participant can compute. Then, \mathcal{C} selects randomly $I, J \in [q_{f_1}]$, $T \in [1, q_s]$, responds to the queries as follows.

- **Create**(ID_i): \mathcal{C} maintains an initially empty list L_c consisting of tuples of the form (ID_i, u_i, X_i) . If $ID_i = ID_I$, \mathcal{C} selects a random $r_i, h_i \in \mathbb{Z}_q^*$ and computes $R_i = r_i P$, $s_i = (e_i + h_i s) \bmod q$, public key $X_i = u_i P$ then i lets partial private key, private key and public key are $Q_i = (s_i, R_i)$, $d_i = \{\perp, Q_i\}$ and i 's public key is X_i . Otherwise, \mathcal{C} selects randomly $u_i, e_i, h_i \in \mathbb{Z}_n^*$ and computes $s_i = e_i + sh_i$, $R_i = e_i P$ and $X_i = u_i P$ separately. Then i 's partial private key, private key and public key are $Q_i = (s_i, R_i)$, $d_i = \{u_i, Q_i\}$ and X_i . Finally, \mathcal{C} adds a tuple (ID_i, R_i, h_i) and (ID_i, Q_i, u_i, X_i) to the list L_{f_1} and L_c , separately. \mathcal{C} answers A_2 's $f_1(ID_i, R_i)$, $\text{Public} - \text{Key}(ID_i)$, $\text{Corrupt}(ID_i)$, $\text{Send}(\Phi_{i,j}^n, \mu)$, $\text{Reveal}(\Phi_{i,j}^n)$, f_2 and $\text{Test}(\Phi_{I,J}^T)$ queries as it is done in Lemma 2. The probability that challenger \mathcal{C} selects $\Phi_{I,J}^T$ as the **Test** oracle is $\frac{1}{q_c^2 q_s}$. In this case, challenger \mathcal{C} would not have made $\text{Corrupt}(\Phi_{I,J}^T)$ or $\text{Reveal}(\Phi_{I,J}^T)$ queries, and so challenger \mathcal{C} would not have aborted. If challenger \mathcal{C} can win in such game, then challenger \mathcal{C} must have made the corresponding f_2 query of the form $(ID_T^i, ID_T^j, T_T^i, T_T^j, K_T^1, K_T^2)$. If $\Phi_{I,J}^T$ is the initiator oracle. Else $(ID_T^i, ID_T^j, T_T^i, T_T^j, K_T^1, K_T^2)$, with overwhelming probability because f_2 is a random oracle. Thus challenger \mathcal{C} can

find the corresponding item in the f_2 -list with the probability $\frac{1}{q_{f_2}}$ and outputs $K_T^1 - s_I bP - r_{I,J}^T(R_J + h_J P_0)$ as a solution to the CDH problem. The probability that \mathcal{C} solves the CDH problem is $\frac{\varepsilon}{q_z^2 q_s q_{f_2}}$.

Theorem 2. *The proposed protocol provides the perfect forward security if the CDH assumption in \mathbb{G} is hard.*

Proof: Assuming that C and node N compute the session key K by applying CLAKA protocol, then after, the private keys SK_C and SK_N get compromised. Assume that a and b are secret values used by PDA and node M when they compute a common session key. For an attacker who possesses SK_C , SK_N , $T_C = aP$ and $T_N = bP$ for secrets a and b , must reveal abP . To reveal the value abP without knowing either a or b , the attacker should be able to solve the CDH problem in \mathbb{G} . Under the CDH, the probability is negligible. Therefore, the CLAKA proves the perfect forward secrecy feature.

6.2 Properties of Our Protocol

In this subsection, we discuss how informal security properties are achieved in the proposed protocol.

- **Unknown key share:** Adversary cannot use the correct private key to encrypt and sign the message. This is considered as the use of unknown key which can not be accepted. This property is satisfied because at each session a new key is established between C and node N , and it is hard to compute $c = ab$.
- **Key compromise impersonation:** If a WBAN controller's long-term key leaks, the adversary will send a request to the Key generation center to query controller's partial private key; then the Type I attack is met. However in our protocol if an adversary wants to find the master key or a private key of an entity, he has to give aP to seek a ; from our assumption of a hard problem on the elliptic curve, of a group G with generator P , for an unknown.
- **Key control:** None of the users can decide to compute the session key because it is derived from a temporary key and computed by two parties C and N .
- **Key escrow:** Since a malicious KGC computes a partial private key Q_i . It does not compute $d_i = (u_i, Q_i)$ because the controller C and node N choose randomly u_i to complete their private keys.
- **Anonymity:** The proposed protocol protects anonymity of nodes during the mediated signature creation. since the content of the message is not revealed.
- **Norepudiation:** Other nodes on blockchain can not deny the use of WBAN data since they can verify the authenticity of node N since its signature is verified.
- **Immutability:** Since the data broadcast by node N forms a blockchain ledger; no other node can modify its content.
- **Revocation:** SEM firstly checks whether the user's identity is revoked and returns error otherwise SEM computes a partial signature as e_s, t .

- **Verifiability:** Blockchain transaction are publicly known to the chain. Any node can check the transactions and hash along way back to the previous block.
- **Consensus mechanism:** A controller C send a consensus message $K||\mu$ to the blockchain as a permission to use its data. This is important before the use of data.

6.3 Comparison with Existing Protocols

Security Comparison. We compare the proposed protocol with the existing authentication and key agreement protocols designed for healthcare systems in Tables 4 and 5 such as Shen et al. [6], Shen et al. [7], Li et al. [8], Wazid et al. [10], and Wazid et al. [11]. The comparison focuses session key, key escrow resiliency, revocability, immutability, consensus, and architecture (decentralized). The comparison shows that our authenticated key agreement for blockchain based WBAN is the best compared to the existing protocols listed in Table 3.

Table 3. Functionality features comparison

Feature	[6]	[7]	[8]	[10]	[11]	Ours
Session key	✓	✓	✓	✓	✓	✓
Key escrow	✓	✓	✓	✗	✗	✓
Revocation	✗	✗	✗	✓	✗	✓
Immutability	✗	✗	✗	✗	✗	✓
Consensus	✗	✗	✗	✗	✗	✓
Decentralized	✗	✗	✗	✗	✗	✓

Computation and Communication Cost. In this section, we compare computation and communication costs of our protocol with existing protocols [7, 8, 11]. The Table 4 contains the number of exchanged messages, point multiplication, and hashes. The computation capabilities of WBANs mobile device are limited. Therefore, we compare the computation and communication (number of exchanged messages) cost of controller/client in our case. For convenience, some notations used in comparison are mentioned as follows.

- T_m : Execution time of a point multiplication
- T_h : Execution time of a hash function.

We refer to the execution time in [19], that was implemented using TinyPairing library on a MICAz mote, that has been used extensively in wireless sensor networks research and possesses only 4 KB RAM, 128 KB ROM, and a 7.3828-MHz ATmega128L microcontroller. In their implementation, the RAM and ROM usage by each operation was obtained using the TinyOS toolchain. We use their

implementation to evaluate the computation cost of the controller/client. For our comparison, an ECC point multiplication is estimated to 0.0171 s while a hash function computation time is 0.00032 s according to Xiong et al. [19]. Therefore, the protocol [11] computes $6T_m + 2T_h = 0.10324$ s, the protocol [8] computes $5T_m + 2T_h = 0.08614$ s, the protocol [7] computes $6T_m + 17T_h = 0.10804$ s, and the proposed protocol computes $6T_m + 2T_h = 0.10324$ s. In Table 4, the protocol [8] has better performance over our proposed protocol but ours adds more security features such as revocation, immutability, verifiability, finality, consensus gained from a security mediated signature and blockchain designed in this paper. The Fig. 2 represents the execution time comparison of our protocol with existing protocols.

Table 4. Computation and Communication costs comparison

Protocol	No. of messages	Computation cost
[11]	4	$6T_m + 2T_h = 0.10324$ s
[8]	2	$5T_m + 2T_h = 0.08614$ s
[7]	3	$6T_m + 17T_h = 0.10804$ s
Ours	2	$6T_m + 2T_h = 0.10324$ s

Note: T_m : Execution time for a point multiplication.
 T_h : Execution time for a hash function.

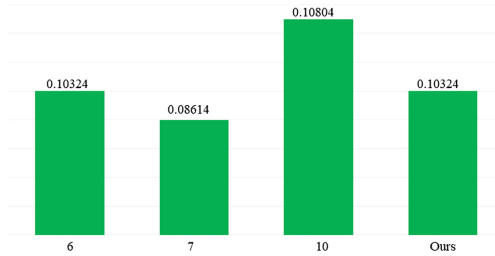


Fig. 2. The computation cost

7 Conclusion

Authenticated key agreement protocols are important for WBANs to provide security and privacy of sensitive information. Thus, a certificateless authenticated key agreement for a decentralized WBAN protocol is proposed. A session key is established between controller C and blockchain nodes N to assure a secure communication. A CLAKA for decentralized WBAN achieves more security features than the existing CLAKA centralized-based WBAN such as immutability, verifiability, consensus, and revocation. In addition to that, a security mediated signature between blockchain nodes is presented to provide instant revocation and verify the eligibility of nodes. The proposed protocol is secure in a random oracle model. It is a lightweight for low capability devices.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (grant no. 61872058).

References

1. Li, F., Hong, J.: Efficient certificateless access control for wireless body area networks. *IEEE Sens. J.* **16**(13), 5389–5396 (2016)
2. Jin, Y.: Low-cost and active control of radiation of wearable medical health device for wireless body area network. *J. Med. Syst.* **43**(5), 137 (2019)
3. Sun, W., Cai, Z., Li, Y., Liu, F., Fang, S., Wang, G.: Security and privacy in the medical internet of things: a review. *Secur. Commun. Netw.* **2018**, (2018)
4. Chen, G., Xu, B., Lu, M., Chen, N.S.: Exploring blockchain technology and its potential applications for education. *Smart Learn. Environ.* **5**(1), 1 (2018)
5. Xu, J.J.: Are blockchains immune to all malicious attacks? *Financ. Innov.* **2**(1), 25 (2016)
6. Shen, J., Chang, S., Shen, J., Liu, Q., Sun, X.: A lightweight multi-layer authentication protocol for wireless body area networks. *Futur. Gener. Comput. Syst.* **78**, 956–963 (2018)
7. Shen, J., Gui, Z., Ji, S., Shen, J., Tan, H., Tang, Y.: Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. *J. Netw. Comput. Appl.* **106**, 117–123 (2018)
8. Li, X., Peng, J., Kumari, S., Wu, F., Karuppiah, M., Choo, K.K.R.: An enhanced 1-round authentication protocol for wireless body area networks with user anonymity. *Comput. Electr. Eng.* **61**, 238–249 (2017)
9. Li, T., Zheng, Y., Zhou, T.: Efficient anonymous authenticated key agreement scheme for wireless body area networks. *Secur. Commun. Netw.* **2017**, 1–8 (2017). <https://doi.org/10.1155/2017/4167549>
10. Wazid, M., Das, A.K., Vasilakos, A.V.: Authenticated key management protocol for cloud-assisted body area sensor networks. *J. Netw. Comput. Appl.* **123**, 112–126 (2018)
11. Wazid, M., Das, A.K., Kumar, N., Conti, M., Vasilakos, A.V.: A novel authentication and key agreement scheme for implantable medical devices deployment. *IEEE J. Biomed. Health Inform.* **22**(4), 1299–1309 (2018)
12. Hankerson, D., Menezes, A.J., Vanstone, S.: Guide to elliptic curve cryptography. *Comput. Rev.* **46**(1), 13 (2005)
13. Oh, J.H., Lee, K.K., Moon, S.J.: How to solve key escrow and identity revocation in identity-based encryption schemes. In: Jajodia, S., Mazumdar, C. (eds.) *ICISS 2005*. LNCS, vol. 3803, pp. 290–303. Springer, Heidelberg (2005). https://doi.org/10.1007/11593980_22
14. Yap, W.-S., Chow, S.S.M., Heng, S.-H., Goi, B.-M.: Security mediated certificateless signatures. In: Katz, J., Yung, M. (eds.) *ACNS 2007*. LNCS, vol. 4521, pp. 459–477. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72738-5_30
15. Zhang, L., Zhang, F., Wu, Q., Domingo-Ferrer, J.: Simulatable certificateless two-party authenticated key agreement protocol. *Inf. Sci.* **180**(6), 1020–1030 (2010)
16. He, D., Chen, J., Hu, J.: A pairing-free certificateless authenticated key agreement protocol. *Int. J. Commun. Syst.* **25**(2), 221–230 (2012)
17. He, D., Chen, Y., Chen, J., Zhang, R., Han, W.: A new two-round certificateless authenticated key agreement protocol without bilinear pairings. *Math. Comput. Model.* **54**(11–12), 3143–3152 (2011)

18. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security, pp. 62–73. ACM (1993)
19. Xiong, X., Wong, D.S., Deng, X.: Tinypairing: a fast and lightweight pairing-based cryptographic library for wireless sensor networks. In: 2010 IEEE Wireless Communication and Networking Conference, pp. 1–6. IEEE (2010)

Blockchain



A Certificateless Proxy Re-encryption Scheme for Cloud-Based Blockchain

Nabeil Eltayieb¹, Liang Sun², Ke Wang², and Fagen Li¹(✉)

¹ Center for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
fagenli@uestc.edu.cn

² SI-TECH Information Technology Co., Ltd, Beijing, China

Abstract. Cloud computing is a powerful technology because it provides users with attractive online files sharing services. However, security and privacy are significant challenges since the cloud cannot be fully trusted due the traditional centralized management system. This paper proposes a certificateless proxy re-encryption as an efficient mechanism to secure access over outsourced data. The proposed scheme relies on blockchain technology for decentralized security administration and data protection. Besides, the scheme achieves data confidentiality and efficient revocation mechanism. Moreover, the security analysis proves the confidentiality and integrity of the data stored in the cloud server. Finally, we evaluate the performance of the proposed scheme.

Keywords: Cloud computing · Blockchain technology · Certificateless proxy re-encryption · Confidentiality

1 Introduction

The rapid growth of cloud computing has attracted many users and organizations to store their data. Cloud server affords huge computation capacity and big memory space with low price [1]. In spite of the benefits, sending sensitive data to the cloud brings several security threats [2]. To overcome these security concerns, the common way is sending these sensitive data in encrypted form instead of plaintext. Following are the general data security and performance problems in most cloud data storage. (1) A single point of failure: The centralized storage in the cloud brings a higher risk of losing the data. For example, if an intruder attacks the cloud server, the privacy of users will be exposed. (2) Data ownership: When the data owner sent his sensitive data to the cloud server, he loses full control over it. Unauthorized data sharing and distribution is easily occurring due to the possible collusion between the authorized users and the malicious cloud server. (3) Data transparency and auditability: All users do not have full transparency over what data is being gathered about them and how they are reached. (4) Computation and communication overhead: Any modification in the access control policy requires more operations on both the data owner and cloud server.

To better protect the data confidentiality, we should move data storage and sharing of the data from centralized to decentralized storage systems. Consequently, it is strongly desirable to design a new data access control scheme to achieve secure data sharing and flexible revocation. To address these problems, many researchers [3–5] started using blockchain technology for the data access control. The blockchain is a public, decentralized, Byzantine fault-tolerant, and immutable ledger, where records are added in temporal sequence [6]. The motivation is to leverage the blockchain technology in the cloud for encryption-based access control and key management. This paper proposes a data access control scheme for the decentralized cloud by combining the decentralized storage system with the blockchain and certificateless proxy re-encryption (CLPRE) technology. Our contributions in this paper are noted below.

1. We introduce a Certificateless Proxy Re-encryption scheme for Cloud-based Blockchain (CPRCB) to secure data access control.
2. Utilizing the blockchain infrastructures in cloud server provides full data transparency and auditability, besides reducing the need for trust third party.
3. The proposed scheme can provide data confidentiality, decentralization, and secure revocation mechanism by updating the users' keys.

2 Related Work

2.1 Certificateless Public Key Cryptography

In order to issue digital certificates that bind data users to their public keys, the traditional public key algorithms need a trusted Certificate Authority (CA). However, certificate management is very costly and complicated because the CA has to create its signature on each user's public key and control each user's certificate. To address such weakness, Boneh and Franklin introduced Identity-based encryption [7], that depends on Key Generation Center (KGC) which generates the private keys of all users leading to the key escrow problem. To add more restriction on data access control, Bethencourt et al. [8] proposed Ciphertext-Attribute-Based Encryption (CP-ABE). The data owner encrypted their data according to the access policy, which is built based on the user's attributes. But, ABE is suffering from revocation problem since the private keys are granted to current users should be updated whenever a user is revoked. In order to solve all the problem mentioned above, Certificateless Public Key Cryptography (CL-PKC) was introduced by Al-Riyami and Paterson [9]. According to their scheme, various certificateless public key encryption (CL-PKE) protocols were introduced in [10–12]. Then, for securing data access in an untrusted cloud server, Lei et al. [13] proposed the CL-PRE (Certificateless Proxy Re-Encryption) protocol.

2.2 Blockchain Technology

Blockchain technology [14] has attracted many users in both academia and industry, which solve most of the complicated address administration and security concerns in distributed systems. Nowadays, blockchain is used to give more strong

access control, which enforces several fields. For example, cloud computing [15], Internet of Things [16], electronic voting [17, 18], smart cities [19], and healthcare [20]. A blockchain is a kind of distributed database; it consists of blocks which are cryptographically linked. Each block is connected with the previous block with a hash value generated by using the SHA-256 algorithm (see Fig. 1).

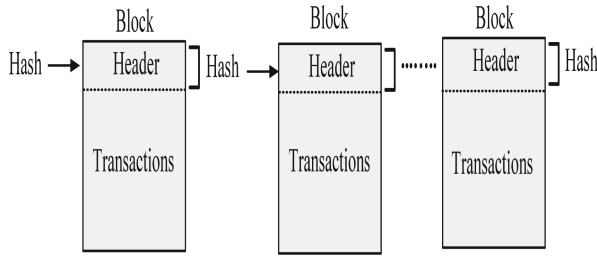


Fig. 1. Blockchain structure

Blockchain workflow is shown in Fig. 2. The user generates a transaction which is signed with his private key and deploys to the blockchain network. When any node received this transaction; first, it validates and verifies the authenticity of user A. The transaction is rejected if validation fails. Otherwise, it is collected with pending transactions from the transaction pool and a block is created. When the network reaches the consensus, the new block becomes a part of the blockchain.

3 Preliminaries

3.1 Bilinear Map

Let \mathbb{G}_1 and \mathbb{G}_2 be two cyclic groups of prime order p . g is a generator of \mathbb{G}_1 . $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear map with properties:

1. Bilinearity: $e(g^a, g^b) = e(g, g)^{ab}$ for all $g, g \in \mathbb{G}_1, a, b \in \mathbb{Z}_p^*$.
2. Non-degeneracy: $g \in \mathbb{G}_1$ satisfy $e(g, g) \neq 1$, where 1 is an identity element in \mathbb{G}_2 .
3. Computability: There is algorithm to compute $e(g, g)$ for all $g, g \in \mathbb{G}_1$.

3.2 The Decisional Bilinear Diffie Hellman (DBDH) Assumption

Given $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, and $g \in \mathbb{G}_1$. The DBDH problem is to determine if given $(g^a, g^b, g^c, T) \in \mathbb{G}_1 \times \mathbb{G}_2$, where $a, b, c \in \mathbb{Z}_p^*, T = e(g, g)^{abc}$ or $T \in \mathbb{G}_2$.

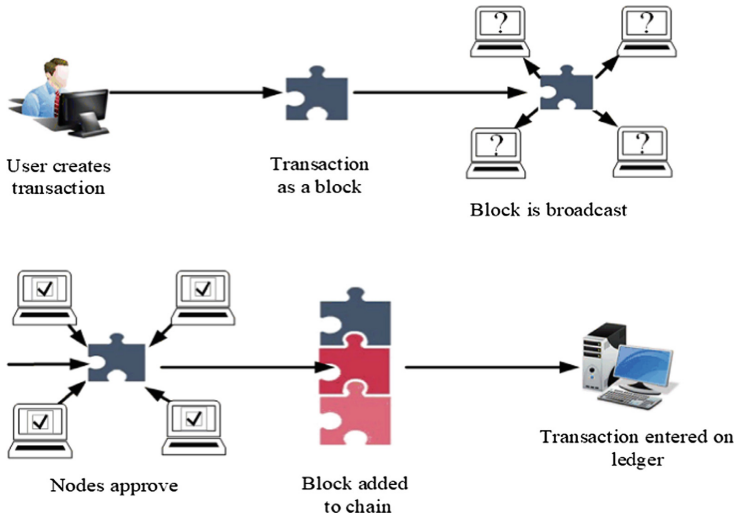


Fig. 2. Blockchain workflow

3.3 Proxy Re-encryption:

Proxy re-encryption is an efficient approach which enables a trusted server to transform a ciphertext under one key into the same ciphertext under different keys of the receivers, without leaking the information of the data (see Fig. 3). The first who introduced this concept is Blaze et al. [21]. A series of proxy re-encryption schemes are added by Ateniese et al. [22]. Proxy re-encryption is mostly utilized in several fields such as intellectual property protection [23], distributed file administration [24], and mail filter. A user U_1 encrypts the data using his public key. When he needs to share the data with another user U_2 , he transfers the ciphertext to a proxy server. The proxy server then reforms the data encrypted under U_1 's public key into data that is encrypted under U_2 's public key and gives this to U_2 . Now U_2 can use his private key to decrypt the ciphertext and expose the contents.

4 The Overview of CPRCB

4.1 System Model

Fig. 4 is presented the model of CPRCB, which involves the following entities:

1. Data Owner (DO). DO is the entity whose data are to be shared with other data users in the cloud. To ensure confidentiality of his data, DO encrypts data before transmitting to the cloud. Beside the encrypted data, the DO sends an access control list (ACL) indicating the data user group.
2. Data User (DU). The DU is the entity who accesses the DO's data. Both the DO and DUs should be registered on the blockchain.

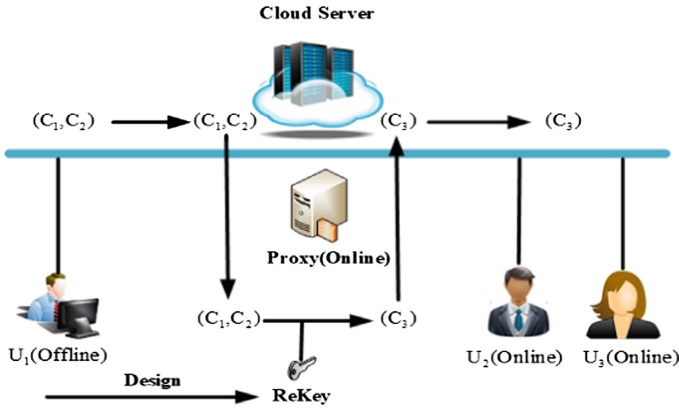


Fig. 3. Proxy re-encryption

3. Cloud Server (CS). It is a repository for the data from DO. The CS stores all encrypted data and controls access based on ACL. To remove the trust from the third party, we consider a cloud-based blockchain which consists of the following entities:
 - (a) Issuer: It is responsible for registering the participants (DO and DU) on the blockchain network. It grants out membership keys to them, and that serves as their identity (ID).
 - (b) Verifier: In the cloud, it represents the authentication unit, responsible for checking whether a user who makes an access request to the data is an authorized user or not.
 - (c) Processing Node: It represents the heart of blockchain network. It performs all the transactions that occur in the system. Moreover, it works as a proxy server that oversees the re-encryption process.
 - (d) Smart Contract Unit: In addition to the access control list created by DO to specify the authorized users. This unit generates a smart contract that states how data are to be used.

4.2 Definition of CPRCB

The proposed CPRCB scheme consists of the following algorithms

1. **Setup** (λ): It takes security parameter λ , broadcasts public parameters $param$, and keeps the master key msk secret.
2. **PPKeyExt**: This algorithm is run by Private Key Generator (PKG) upon input $param$ and the identity of user (ID). It generates the partial private key D .
3. **SKeyGen**: It is run by the user that takes $param$ and D as inputs. In the end, it outputs the user's private key sk .
4. **PkeyGen**: The algorithm is run by the user which takes $param$ as inputs. It outputs the user's public key pk .

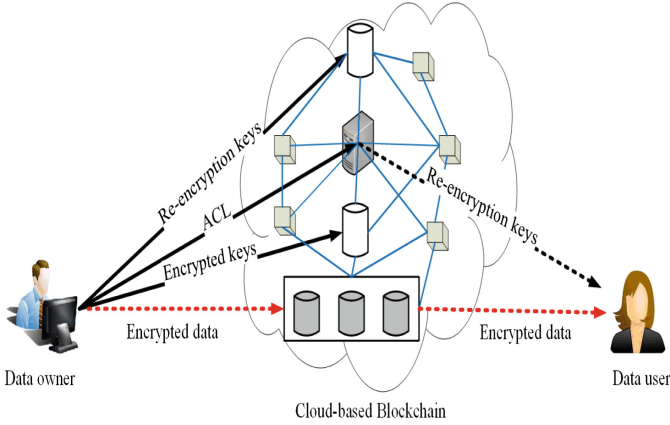


Fig. 4. Network model

5. **Encrypt** The algorithm is executed by the user that takes $param$, pk , and the message m as inputs. It outputs $C(m)$.
6. **Decrypt1**: It is run by the user which takes $param$, sk , and the encrypted message $C(m)$ as inputs. It outputs m .
7. **PREKeyGen**: The data owner runs this algorithm which takes $param$, the public key of user pk_U , and the user's identity ID . It creates the proxy re-encryption key $rk_{O \rightarrow U}$.
8. **ReEnc**: It is called by the proxy re-encryption. It takes as input the $rk_{O \rightarrow U}$ and the ciphertext $C_O(m)$. The algorithm generates the last ciphertext CT , which sends to data user.
9. **Decrypt2**: This algorithm is run by the data user. Upon input $param$, CT , and th user private key sk_U . It outputs m .

4.3 Security Model

Assuming that an adversary \mathcal{A} can obtain a partial private key or the private key of DU, or both. As CPRCB have a proxy re-encryption scheme, \mathcal{A} executes both re-encryption algorithm and re-encryption key generation oracle. We consider chosen plaintext attack (CPA) game between \mathcal{A} and challenger \mathcal{C} as follows.

1. **Setup**. The security parameter λ is taken as input. The challenger \mathcal{C} calls the **Setup** algorithm to produce msk and $param$. msk is kept secret by \mathcal{C} while $param$ is sent to \mathcal{A} .
2. **Phase 1**. The \mathcal{A} asks for the following queries:
 - (a) PPKeyExt query: Upon getting ID , \mathcal{C} executes **PPKeyExt** algorithm to create a partial private key D_{ID} which is transferred to \mathcal{A} .
 - (b) SKeyGen query: Upon receiving ID , \mathcal{C} runs **SKeyGen** to generate a private key sk_{ID} which is sent to \mathcal{A} .
 - (c) PkeyGen query: Upon receiving ID , \mathcal{C} calls **PKeyGen** to generate a private key pk_{ID} which is returned to \mathcal{A} .

- (d) **PREKeyGen** query: \mathcal{C} runs **PREKeyGen** to create the proxy re-encryption key $rk_{ID_1 \rightarrow ID_2}$ and returns it to \mathcal{A} .
- 3. **Challenge.** Pair of messages m_0, m_1 with equal length and identity ID^* are suggested by \mathcal{A} to be challenged. \mathcal{C} replies to \mathcal{A} with the challenge ciphertext CT^*
- 4. **Phase 2.** Similar to **Phase 1**.
- 5. **Guess.** A guess $i' \in \{0, 1\}$ upon i is sent by \mathcal{A} . If $i' = i$, \mathcal{A} wins game. The advantage of adversary \mathcal{A} is indicated as:

$$Adv(\mathcal{A}) = | \Pr[i' = i] - 1/2 | .$$

Definition 1. *CLE-BAC scheme is secure against CPA, if no probabilistic polynomial time adversary \mathcal{A} has a non-negligible advantage in breaking the CLE-BAC.*

4.4 Security Requirements

The proposed scheme satisfies the following security requirements.

- 1. **Data Confidentiality:** The encrypted data should be accessible by the authorized users and at the same time, unauthorized users are prevented from access. Hence, the access control policy should be defined by the data owner before outsourcing the data.
- 2. **Decentralization:** Using the blockchain technology protects the scheme from a single point of failure. For any change in one block; one needs to change every subsequent block before any new block could be mined.
- 3. **Revocation Mechanism:** It is essential for any encryption schemes that involve several users since some private keys might get compromised at some point.

5 The Proposed Scheme

In this section, the concrete construction of our algorithms and the revocation mechanism are given.

5.1 Concrete Construction

This subsection presents the concrete construction of CPRCB, whose main elements are described as follows.

- 1. **Setup (λ):** This algorithm is run by PKG, given λ as a security parameter, $\mathbb{G}_1, \mathbb{G}_2$ two groups of prime order p , and the bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Next, it acts as follows:
 - (a) Picks a group generator $g \in \mathbb{G}_1$.
 - (b) Sets the master key (msk) by selecting an integer $s \in \mathbb{Z}_p^*$ randomly. Then, it broadcasts the public parameters ($param$) as $(\mathbb{G}_1, \mathbb{G}_2, H_1, H_2, g, g^a)$.
 - (c) Chooses hash functions $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$ and $H_2 : \mathbb{G}_2 \rightarrow \mathbb{G}_1$.

2. **PPKeyExt**($param, ID$): To get partial private key, DO sends requests with his identity ID_O to PKG. PKG computes $g_O = H_1(ID_O)$, $D_O = g_O^s$ and transfers D_O to DO. A similar process is done for DU.
3. **SKeyGen**($param, D_O$): DO picks an integer $x_O \in \mathbb{Z}_p^*$ randomly. The same with DU. Then, DO calculates private key:

$$sk_O = D_O^{x_O} = g_O^{s \cdot x_O} \quad (1)$$

A similar process is performed by DU. For delegation purpose DO selects an integer $t \in \mathbb{Z}_p^*$ randomly and it keeps sk_O and t secret.

4. **PkeyGen**($param$): DO calculates his public key:

$$pk_O = (g_O, g^{s \cdot x_O}) \quad (2)$$

where $g_O = H_1(ID_O)$. DO publishes pk_O to let users send him messages. For decryption delegation DO publishes g^t . A similar process is done for DU. Note, DU can calculate his public key without the sending to PKG.

5. **Encrypt**($param, m, pk_O$): In order to encrypt a message $m \in \mathbb{G}_2$ that can only be decrypted by himself, DO selects an integer r and computes

$$c = C_O(m) = (g^r, m \cdot e(g_O^r, g^{s \cdot x_O}))$$

For decryption delegation purpose DO selects an integer r and computes

$$c' = C'_O(m) = (g^{tr}, g^r, m \cdot e(g_O^r, g^{s \cdot x_O}))$$

6. **Decrypt1**($param, C_O(m), sk_O$): To decrypt $C_O(m) = (u, v)$ using sk_O , DO calculates

$$v/e(sk_O, u) = m \cdot e(g_O, g^{s \cdot x_O})^r / e(g^{s \cdot x_O}, g^r) = m$$

7. **PREKeyGen**($param, ID_U, pk_U$): In order to delegate decryption right to DU, DO picks $x \in \mathbb{G}_2$ randomly and calculates proxy re-encryption key as:

$$rk_{O \rightarrow U} = (g_O^{-s \cdot x_O} \cdot H_2^t(x), C_U(x)) \quad (3)$$

In the end, DO sends this key to the proxy.

8. **ReEnc**($rk_{O \rightarrow U}, C'_O(m)$): To re-encrypt a ciphertext $C'_O(m)$ by using the re-encryption key $rk_{O \rightarrow U}$, the proxy calculates

$$\begin{aligned} c'' &= m \cdot (g_O, g^{s \cdot x_O})^r \cdot e(g_O^{-s \cdot x_O} \cdot H_2^t(x), g^r) \\ &= m \cdot e(H_2^t(x), g^r), \end{aligned}$$

and then transfers $CT = (g^{tr}, c'', C_U(x))$ to DU.

9. **Decrypt2**($param, CT, sk_U$): When DU received CT , firstly, it decrypts $C_U(x)$ to retrieve x and then recovers m by computing

$$c'' / e(H_2(x), g^{tr}) = m.$$

5.2 Revocation Mechanism

Revocation mechanism is vital for securing data access control in cloud computing. One efficient way for the revocation used in our scheme is key updating for existing re-encryption keys.

If Do needs to update his keys (public/private), he selects $x'_O \in_R \mathbb{Z}_p^*$, sets the new private key as $sk'_O = D^{x'_O} = g^{s \cdot x'_O}$ and new public key as $pk'_O = (g_O, g^{s \cdot x'_O}, g^{t \cdot x'_O / x_O})$. To employ the cloud to update associated re-encryption keys, DO computes a number $r_{O' / O} = x'_O / x_O \bmod p$ and transfers to the proxy. When the proxy receiving $r_{O' / O}$, it updates associated re-encryption keys $rk_{O \rightarrow U}$ as follows:

$$\begin{aligned} rk'_{O \rightarrow U} &= (g_O^{-s \cdot x_O} \cdot H_2(x)^{t \cdot r_{O' / O}}, C_U(x)) \\ &= (g_O^{-s \cdot x_O} \cdot H_2(x)^{t \cdot x'_O / x_O}, C_U(x)) \end{aligned}$$

Do should save the new list of the secret values. To check the correctness of the updated re-encryption key. Suppose that DO encrypts a message under his new public key as

$$(g^{tr \cdot x'_O / x_O}, g^r \cdot m \cdot e(g_O^r, g^{s \cdot x'_O}))$$

For re-encryption, the proxy server calculates

$$(m \cdot e(g_O^r, g^{s \cdot x'_O})) \cdot e(g^{s \cdot x'_O} \cdot H_2(x)^{t \cdot x'_O / x_O}, g^r) = m \cdot e(H_2(x)^{t \cdot x'_O / x_O}, g^r)$$

Finally, the DU can recover the message by computing

$$m \cdot e(H_2(x)^{t \cdot x'_O / x_O}, g^r) / e(H_2(x), g^{tr \cdot x'_O / x_O}) = m$$

6 Security Analysis and Discussion

6.1 Security Proof

Theorem 1. *The proposed CPRCB scheme is CPA secure in the random oracle model under the DBDH assumption.*

Proof. PPT algorithm \mathcal{B} is considered to solve DBDH problem. Then, \mathcal{B} accepts the tuple a, b, c, T as the instance of the DBDH problem as presented in Fig. 5. It outputs 1 if $T = e(g, g)^{abc}$. \mathcal{B} plays the role of the random oracle and simulates $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$ as follows:

- If ID has not been queried before, \mathcal{B} selects $x, z, t \in \mathbb{Z}_p^*$. Then, it flips a coin to set $\alpha \leftarrow 1$ with probability γ ; otherwise, sets $\alpha \leftarrow 0$.
- If $\alpha \leftarrow 0$, set $h \leftarrow (g^c)^z$, else calculate $h \leftarrow g^z$.
- \mathcal{B} saves the tuple (ID, h, x, z, t, α) and answers with h as the results.

Following are simulation process of the CPA game between \mathcal{B} and \mathcal{A} .

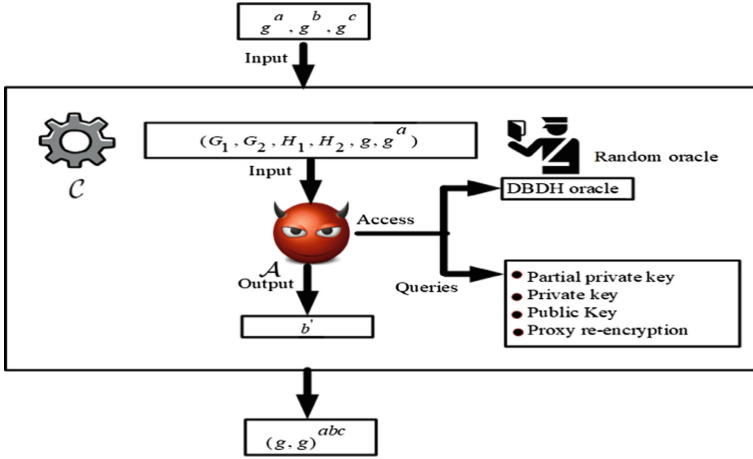


Fig. 5. Security proof diagram of CPA

1. **Setup.** The public parameters $(\mathbb{G}_1, \mathbb{G}_2, H_1, H_2, g, g^a)$ are created by \mathcal{B} and sent to \mathcal{A} . a represents the master key of PKG.
2. **Phase 1.** \mathcal{A} asks for the following queries:
 - (a) PPKeyExt Query: \mathcal{A} asks for partial private key. \mathcal{B} runs **PPKeyExt**(ID) algorithm, and evaluates $H_1(ID)$ to get (ID, h, x, z, t, α) , and transfers $(g^a)^z$ to \mathcal{A} .
 - (b) SKeyGen Query: When \mathcal{A} sends query for private key, \mathcal{B} runs **SKeyGen**(ID) algorithm and evaluates $H_1(ID)$ to get (ID, h, x, z, t, α) and returns $((g^a)^z)^x$ to \mathcal{A} .
 - (c) PkeyGen Query: When \mathcal{A} sends query for public key, \mathcal{B} runs **PKeyGen**(ID) algorithm and evaluates $H_1(ID)$ to get (ID, h, x, z, t, α) and returns $((g^a)^z)^x$ to \mathcal{A} .
 - (d) PREKeyGen Query: When \mathcal{A} sends query for proxy re-encryption key, \mathcal{B} runs **PREKeyGen**(ID_1, ID_2) algorithm. \mathcal{B} evaluates $H_1(ID_1)$ and $H_1(ID_2)$ to get $(ID_1, h_1, x_1, z_1, t_1, \alpha_1)$ and $(ID_1, h_2, x_2, z_2, t_2, \alpha_2)$. Also, \mathcal{B} selects $r \xleftarrow{R} \mathbb{Z}_p^*$, $x \xleftarrow{R} \mathbb{G}_1$ and $X \xleftarrow{R} \mathbb{G}_2$. If $\alpha_1 = 0$, \mathcal{B} returns proxy re-encryption key to \mathcal{A} as:

$$rk_{ID_1 \rightarrow ID_2} = (x, (g^b)^r, X.T^{rz_2x_2})$$

If $\alpha_1 = 1$, \mathcal{B} returns proxy re-encryption key to \mathcal{A} as:

$$rk_{ID_1 \rightarrow ID_2} = (x, (g^a)^{-z_1x_1}, H_2(X^{t_1}), C_{ID_2}(X))$$

3. **Challenge.** \mathcal{A} selects (ID^*, m_0, m_1) , ID^* should not be trivial. \mathcal{A} obtains his private key for ID^* after \mathcal{B} run **SKeyGen**(ID_*) algorithm. \mathcal{A} obtains the proxy re-encryption key from ID^* to ID' . Then, \mathcal{B} evaluates $H_1(ID_*)$ and gets $(ID^*, h, x, z, t, \alpha)$. Next, \mathcal{B} selects $i \in_R 0, 1$ and sends $(g^b, m_i.T^{zx})$ to \mathcal{A} .

4. **Phase 2.** Similar to **Phase 1**, a trivial query is not allowed.
5. **Guess.** The \mathcal{A} outputs a guess bit i' . The value of α created by $H_1(ID_i)$ is α_i . The following conditions are checked by \mathcal{B} .
 - (a) The value α corresponding to ID^* is 0.
 - (b) For all \mathcal{A} 's queries **SKeyGen**(ID_i) and **PKeyGen**(ID_i), $\alpha_i = 1$.
 - (c) For all \mathcal{A} 's queries **PREKeyGen**(ID_i, ID_j), where $ID_i \rightarrow ID_j$ is created with the same method to ID^* , $\alpha_j = 0$.
 - (d) For all \mathcal{A} 's queries **PREKeyGen**(ID_i, ID_j), where $ID_i \rightarrow ID_j$ is not created with the same method to ID^* , $\alpha_j = 1$.

If any of these conditions untrue, \mathcal{B} exits the game. If \mathcal{B} does not exit, it outputs 1 if $i' = i$, otherwise output 0. The challenge ciphertext CT^* is a correct encryption of m_i under ID^* and hence $Adv(\mathcal{A}) = \left| \Pr[i' = i] - 1/2 \right| \geq \epsilon$.

6.2 Discussion

Using blockchain technology in cloud computing servers to secure data sharing provides additional restrictions and unchanging log of all significant security events. These benefits are made possible by the following features:

1. **Decentralization:** The information is equally distributed between the nodes. The public validation of each transaction allows anyone to verify if the system is working correctly, using the distributed ledger records. Furthermore, the decentralization protects the scheme from a single point of failure. For any change in one block; one needs to change every subsequent block before any new block could be mined.
2. **Cryptography:** The structure of blockchain is strong due to the cryptographic hash techniques applied. Hash values are used to hide true identities. Moreover, this hashing value is created using the SHA-256 algorithm to map data of arbitrary size to data with a fixed size.
3. **Consensus:** It determines which node can add a block after that node is the winner of the cryptographic race. This kind of consensus is defined as proof of work. It assures each block has passed complex mathematical operations before becoming an immutable part of the blockchain.

7 Performance Evaluation

In Table 1, the proposed scheme and the schemes in [25–28] are compared in terms of computation cost and properties. We get the running time for the cryptographic operations using Pairing-Based Cryptography (PBC) [29]. The experiment is carried out on a PC (i5-7400, 3 GHz processor with 8-GB Ram and a Windows 10- 64-bit) using VC++ 6.0. The running times of one pairing operation and one exponentiation operation are 13.455 ms, 6.441 ms, respectively.

7.1 Computation Cost

Table 1 and Fig. 6 show the computation overhead of CPRCB scheme and other schemes. The detailed analysis is shown as follow.

1. Encryption: The time cost for the encryption algorithm in CPRCB is 26.327 ms which is slightly greater than the time cost in [25] which costs 26.327 ms. While the costs of [26–28] are 32.768 ms, 38.646 ms, 39.772, respectively.
2. Decryption: The time cost for the decryption algorithm in CPRCB is 13.455 ms, which approximately equals the time costs in [25] and [27], which cost 12.882, 12.882, respectively.

Table 1. Comparison.

		[25]	[26]	[27]	[28]	CPRCB
Computation cost	Enc	4exp	3exp+1pair	6exp	2exp+2pair	2exp+1pair
	Dec	2exp	4exp+1pair	2exp	2pair	1pair
Properties	Data confidentiality	✓	✓	✓	✓	✓
	Decentralization	×	×	×	×	✓
	Revocation	×	×	✓	×	✓

Legends: Enc and Dec: the encryption and the decryption, respectively; exp: exponentiation operation; Pair: pairing operations; Other operations are ignored; ×: this method is not used in the corresponding scheme; ✓: this method is used in the corresponding scheme

7.2 Properties

1. Data confidentiality: For confidentiality, unauthorized user and the cloud server have no access the plaintext. Theorem 1 demonstrated that the proposed scheme is secure against chosen plaintext attack (CPA).
2. Decentralization: The cloud-based blockchain is more difficult to break than traditional centralized data. The decentralized nature of the blockchain combined with digitally-signed transactions ensures that an adversary cannot impersonate an authorized user or tamper the network. Besides, the decentralization protects the scheme from a single point of failure.
3. Revocation mechanism: The data owner can only revoke a specified user based on his identity by updating the re-encryption keys.

Compared our scheme and the scheme in [25–28], we noticed that CPRCB achieved data confidentiality, decentralization, and secure revocation.

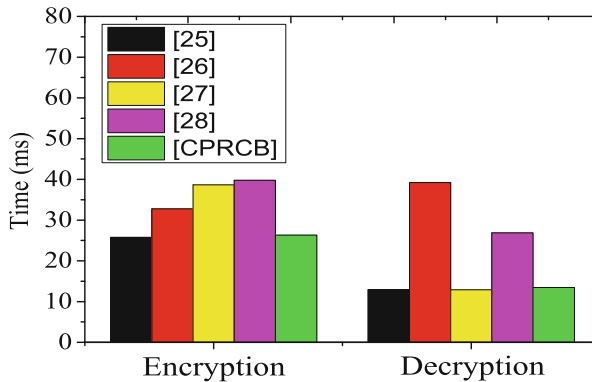


Fig. 6. Computation overhead

8 Conclusion

In this paper, a certificateless proxy re-encryption for cloud-based blockchain is presented to secure access over outsourced data. The proposed scheme relies on blockchain technology for decentralized security administration and data protection. Besides, the scheme provides data confidentiality and efficient revocation mechanism. Moreover, the security analysis has shown that our scheme is able to prevent a chosen plaintext attack (CPA). The performance evaluation showed that the proposed scheme is efficient as far as the computation cost and security properties. As future work, we will focus on the deployment of smart contracts on Ethereum.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (grant no. 61872058).

References

1. Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M.: A break in the clouds: towards a cloud definition. *ACM SIGCOMM Comput. Commun. Rev.* **39**(1), 50–55 (2008)
2. Jensen, M., Schwenk, J., Gruschka, N., Iacono, L.L.: On technical security issues in cloud computing. In: 2009 IEEE International Conference on Cloud Computing, pp. 109–116. IEEE (2009)
3. Ouaddah, A., Abou Elkalam, A., Ait Ouahman, A.: Fairaccess: a new blockchain-based access control framework for the internet of things. *Secur. Commun. Netw.* **9**(18), 5943–5964 (2016)
4. Ouaddah, A., Elkalam, A.A., Ouahman, A.A.: Towards a novel privacy-preserving access control model based on blockchain technology in iot. *Europe and MENA Cooperation Advances in Information and Communication Technologies*, vol. 520. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-46568-5_53

5. Zyskind, G., Nathan, O., et al.: Decentralizing privacy: using blockchain to protect personal data. In: 2015 IEEE Security and Privacy Workshops, pp. 180–184. IEEE (2015)
6. Swan, M.: Blockchain: Blueprint for a New Economy. O'Reilly Media Inc, Newton (2015)
7. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
8. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP 2007), pp. 321–334. IEEE (2007)
9. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Lai, 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
10. Elhabob, R., Zhao, Y., Sella, I., Xiong, H.: Efficient certificateless public key cryptography with equality test for internet of vehicles. In: IEEE Access (2019)
11. Li, F., Hong, J., Omala, A.A.: Efficient certificateless access control for industrial internet of things. *Future Gener. Comput. Syst.* **76**, 285–292 (2017)
12. Hassan, A., Eltayieb, N., Elhabob, R., Li, F.: A provably secure certificateless user authentication protocol for mobile client-server environment. *Advances in Internetworking, Data and Web Technologies*, vol. 6. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-59463-7_59
13. Xu, L., Wu, X., Zhang, X.: Cl-pre: a certificateless proxy re-encryption scheme for secure data sharing with public cloud. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, pp. 87–88. ACM (2012)
14. Nakamoto, S., et al.: Bitcoin: a peer-to-peer electronic cash system (2008)
15. Dong, C., Wang, Y., Aldweesh, A., McCorry, P., van Moorsel, A.: Betrayal, distrust, and rationality: smart counter-collusion contracts for verifiable cloud computing. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 211–227. ACM (2017)
16. Huang, H., Chen, X., Wu, Q., Huang, X., Shen, J.: Bitcoin-based fair payments for outsourcing computations of fog devices. *Future Gener. Comput. Syst.* **78**, 850–858 (2018)
17. McCorry, P., Shahandashti, S.F., Hao, F.: A smart contract for boardroom voting with maximum voter privacy. In: Kiayias, A. (ed.) FC 2017. LNCS, vol. 10322, pp. 357–375. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70972-7_20
18. Wang, Z., Zhang, H., Song, X., Zhang, H.: Consensus problems for discrete-time agents with communication delay. *Int. J. Control Autom. Syst.* **15**(4), 1515–1523 (2017)
19. Reilly, E., Maloney, M., Siegel, M., Falco, G.: A smart city IOT integrity-first communication protocol via an ethereum blockchain light client. In: Proceedings of the International Workshop on Software Engineering Research and Practices for the Internet of Things (SERP4IoT 2019), Marrakech, Morocco, pp. 15–19 (2019)
20. Yaeger, K., Martini, M., Rasouli, J., Costa, A.: Emerging blockchain technology solutions for modern healthcare infrastructure. *J. Sci. Innov. Med.* **2**(1) (2019)
21. 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>
22. Ateniese, G., Fu, K., Green, M., Hohenberger, S.: Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **9**(1), 1–30 (2006)

23. Ibraimi, L., Tang, Q., Hartel, P., Jonker, W.: A type-and-identity-based proxy re-encryption scheme and its application in healthcare. In: Jonker, W., Petković, M. (eds.) *SDM 2008*. LNCS, vol. 5159, pp. 185–198. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85259-9_12
24. Taban, G., Cárdenas, A.A., Gligor, V.D.: Towards a secure and interoperable DRM architecture. In: *Proceedings of the ACM Workshop on Digital Rights Management*, pp. 69–78. ACM (2006)
25. Lin, X.J., Sun, L., Qu, H.: An efficient RSA-based certificateless public key encryption scheme. *Dis. Appl. Math.* **241**, 39–47 (2018)
26. Guo, H., Zhang, Z., Zhang, J., Chen, C.: Towards a secure certificateless proxy re-encryption scheme. In: Susilo, W., Reyhanitabar, R. (eds.) *ProvSec 2013*. LNCS, vol. 8209, pp. 330–346. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41227-1_19
27. Seo, S.H., Nabeel, M., Ding, X., Bertino, E.: An efficient certificateless encryption for secure data sharing in public clouds. *IEEE Trans. Knowl. Data Eng.* **26**(9), 2107–2119 (2013)
28. Sun, Y., Li, H.: Short-ciphertext and bdh-based CCA2 secure certificateless encryption. *Sci. China Inf. Sci.* **53**(10), 2005–2015 (2010)
29. Lynn, B., et al.: Pbc: the pairing-based cryptography library. <http://crypto.stanford.edu/abc>



A Novel Fair and Verifiable Data Trading Scheme

Haiyong Yu¹, Juntao Gao^{1(✉)}, Tong Wu¹, and Xuelian Li²

¹ ISN State Key Laboratory China, Xi'an, Shaanxi, P. R. China
jtgao@mail.xidian.edu.cn

² School of Mathematics and Statistics,
Xidian University, Xi'an, Shaanxi, China

Abstract. With the widespread use of smart devices, a huge volume of data is generated every day, which is helpful for device user and device enterprises. However, the data generated by the smart device contains the user's privacy, and the data is easy to be modified, forged, which requires a suitable scheme to protect the privacy of the data seller, the authenticity of the data, the fairness during the data trading process. In order to solve the problems, we design a novel fair and verifiable data trading scheme by combining hash function, signature, oblivious transfer, smart contract and private blockchain. The hash function is used for data integrity, the signature is used for the source of the data, the oblivious transfer is used for data verification, the smart contract is used for the encryption key trading, and the private blockchain is used as a ledger for the verification record, trading record and user reputation. The performance analysis shows that our scheme has enough features to help users complete data trading, and our scheme provides an extra function, the reputation record of users to reduce the possibility of user being deceived. The security analysis shows that our scheme provides IND-CCA security, anonymity, and has the capability of resisting collusion attack and data seller fraud. The fairness and practicability of the scheme are verified by simulation.

Keywords: Blockchain · Oblivious transfer · Smart contract · Data trading

1 Introduction

According to incomplete statistics, the number of IoT devices in the world in 2015 was only 3.8 billion. By the end of 2018, the number of connected devices has exceeded 17 billion worldwide. Except for connections such as smartphones, tablets, laptops or landlines, the number of IoT devices has reached 7 billion. It predicts that the number of global IoT devices will reach 8.3 billion units in 2019, and the number will exceed 20 billion units by 2025 in addition to the connections of smartphones, tablets, laptops or landlines.

Even though each smart device produces only a small amount of data, such as 10 kb, these smart devices can generate almost 77.3 TB data per day, which is 27.55 PB per year. Everyone can get useful information by analyzing the data related

to them. For example, smart device users can adjust diet and daily habits by analyzing the data of daily calorie intake and exercise situation.

Getting valuable information needs a large amount of data analysis. The data quantity that individual generates cannot support this work. Therefore, users have to get enough data in other ways before data analysis. Getting data through data trading is a great way.

In the data trading scheme, we should firstly guarantee the privacy of the data seller. Secondly, we should ensure that the data purchased from data seller is valid. Thirdly, we also need a flexible solution for users to change roles flexibly during data trading. Finally, we must ensure the fairness and security of users in the data trading process.

In the existing data trading schemes, the processing methods can generally be divided into three types:

Centralized data trading scheme. Similar as traditional trading method, the server is responsible for managing data, verifying data, and managing user identity. Although this method can reduce the complexity of user operations, there are still some problems such as single point of failure and excessive server load pressure.

Semi-distributed data trading scheme. This method differs from the centralized scheme in that the user saves the decryption key and the server is only responsible for storing the ciphertext. The data purchaser trades data with the data seller through a trusted third party who is responsible for verifying the data. However, if the third party is malicious, it will harm the interests of the data seller.

Distributed data transaction scheme. The data purchaser finds the data he needs through the information broadcasted by the data seller. Then directly deals with the data seller. This kind of scheme can greatly reduce the load of the server, but how to ensure the authenticity of data and the fairness of data transaction must be solved.

1.1 Contribution and Organization

In order to solve the four problems that we mentioned earlier, we propose a novel distributed fair and verifiable data trading scheme. We use a distributed system that does not require real identity to ensure the anonymity of users and facilitate users to switch their roles flexibly. Data is verified by oblivious transfer to ensure the authenticity of the data. Encryption key is traded through smart contract to ensure fairness of the trading. Specifically, our contributions are as follows:

- We propose a distributed scheme that does not require the participation of third parties, in which both parties of data transaction are anonymous. Users can sell data as data sellers to gain revenue, or purchase the data they need as data purchasers. We use a private blockchain as a ledger for the user's verification information, transaction information and reputation information, which provides a reference for data purchasers before they purchase data. This not only reduces the risk of users being cheated, but also encourages users to maintain good reputation information for better long-term benefits.
- Oblivious transfer is used for verifying the data. We designed an interactive random number generation scheme to ensure that both parties trading data are satisfied with the random numbers obtained. Utilizing the characteristic of OT random receiving

message ensures that data purchasers can obtain a small amount of data to verify, but not leak too much information about the data itself. The data purchaser verifies the authenticity of the data by decrypting the original text, and determines whether the behavior of the data seller is honest by comparing the hash value of the plaintext and the encryption key.

- Smart contract is used to trade encryption key. We add a process to verify the encryption key in the smart contract, only when the data seller provides the correct encryption key, can he get the currency. Then the data purchaser will get the private key through simple calculations.

The rest of the paper is organized as follows. We introduce some related work in Sect. 2. We present the overview of our scheme in Sect. 3. Section 4 is the details of our scheme. Security, performance and efficiency of the scheme are given in Sect. 5. Finally, Sect. 6 concludes our work.

2 Related Work

Juang et al. proposed a secure digital commodity trading scheme [1] in cloud computing. The scheme provides an effective trading method for buyers and sellers to trade matching digital goods in the cloud. However, using a public key directly encrypting digital content requires high computational overhead, and if there is no valid data verification method, the rights of the data purchaser cannot be well guaranteed.

Based on the premise that digital content is easy to pirate, Chen et al. proposed a complete arbitration mechanism [2] to solve fair transactions between customers and stores. The arbitrator can make a correct judgment. The research focuses on the fairness of the transaction of digital content. The rights of buyers and sellers are guaranteed by trusted third parties.

Hwang et al. proposed an efficient and provable fair document exchange protocol [3] with transaction privacy that allows untrusted buyers and sellers to exchange documents fairly. They also use an arbitrator to ensure the fairness.

Delgado-Segura et al. proposed a fair data transaction protocol [4] based on the Bitcoin scripting language. They use oblivious transfer to verify data, and use the ECDSA vulnerability to trade the private key. If the scheme use the method of symmetric encryption first, it will further reduce the overhead of encryption and decryption.

Kiyomoto et al. introduced the design of a fair trade protocol [5] for anonymous datasets between data agents and data analysts. The scheme uses public key encryption combined with a hash function to ensure the confidentiality and non-tamperability of the data. If using a blockchain as a trusted storage, you must consider the storage cost of the nodes when the amount of data is too large.

Wang et al. proposed a new P2P-based DRM scheme [6] to protect valuable digital content. The scheme reduces the storage overhead of the server by means of p2p, combines symmetric encryption and public key encryption to ensure the confidentiality of digital content, and uses bitcoin transaction scripts to ensure fair transaction of the encryption key.

Zhao et al. proposed a new blockchain-based fair data transaction protocol [7]. They use oblivious transfer and similarity learning to verify data. They use ring signature and two-factor authentication to guarantee user's privacy. And they use Ethereum smart contract to trade the encryption key. But when there is a problem with the transaction, they also use arbitration to ensure the fairness of both parties.

In addition to the above studies, Missier et al. [8] research to achieve the value of data through data trading. Alrawahi et al. [9], Lin et al. [14], Cattelan et al. [15] research to trade data through a platform and design the E-commerce-like protocol. Perera et al. [10], Lin et al. [13] research to encourage people to collect data for data transactions through incentives. Huang et al. [11], Fan et al. [16] research to exchange digital content fairly. Juang et al. [12] research to protect the digital content in cloud computing. Qian et al. [17] research using an offline semi-trusted third party and interactive verification signature to guarantee the security of the data trading.

3 The Overview of Our Scheme

3.1 System Model

As can be seen from Fig. 1, the fair and verifiable data trading scheme consists of four entities: data seller, data purchaser, cloud storage, and private blockchain.

Data Seller (DS): The data seller performs operations such as encryption, signature, etc. to process the data that needs to be sold.

Data Purchaser (DP): The data purchaser can find the data through the data description on the private blockchain. After verifying the data, he purchases the data.

Cloud storage: The cloud storage is responsible for storing ciphertext.

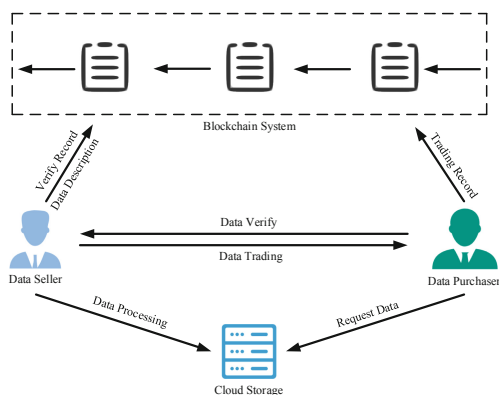


Fig. 1. System Model of the blockchain-based fair data trading scheme.

Private blockchain: The private blockchain stores description information of data, data storage path, the data verification records and the data trading records, and the reputation information of the data seller.

As we described in Fig. 2, our fair and verifiable data trading scheme consists of four phases: system initialization, data processing, data validation, and data purchase. The details of these phases will be described in Sect. 4.

3.2 Introduction to Security Requirements

In our scheme, we show that the RSA encryption provides IND-CCA security. Our scheme can provide confidentiality of the data because we use the symmetric key to encrypt the data, and use the RSA public key to encrypt the symmetric key. Our scheme also provides anonymity by removing the requirements of real identity, prevents single point of failure, and improves the user’s fairness by using the private blockchain.

We will provide detailed proof of safety requirements in Sect. 5.2.

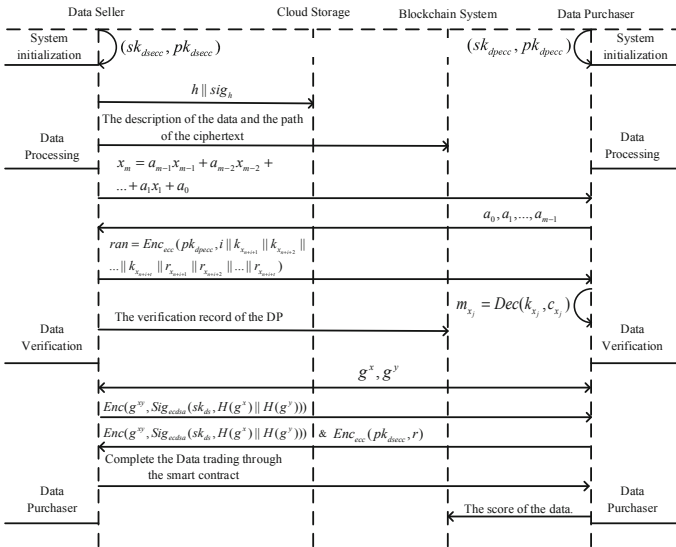


Fig. 2. The framework of the proposed protocol.

4 Our Fair and Verifiable Data Trading Scheme

4.1 System Initialization

1. The user registers in the Ethereum system to obtain the public key and private key (pk_{ecc}, sk_{ecc}) of the Ethereum wallet address.
2. The user registers on the private blockchain, obtains the public key and private key pair (pk'_{ecc}, sk'_{ecc}) . Then the user uses the obtained private key to sign the Ethereum wallet address public key, and records his public key of the Ethereum wallet address on the private blockchain.

3. If the user needs to purchase data, the user can view the data description information through the private blockchain to find the data that he needs.
4. If the user needs to sell the data, the user encrypts and signs the data, upload the data ciphertext and signature to the cloud storage, and records the data description information on the private blockchain.

4.2 Data Processing

When a user wants to sell data to earn money, in order to ensure the confidentiality of his data, he needs to encrypt the data before the data will be sold. At the same time, due to the characteristics of easy modifying, he also needs to calculate the hash value of the plaintext of the data and the symmetric keys (Fig. 3).

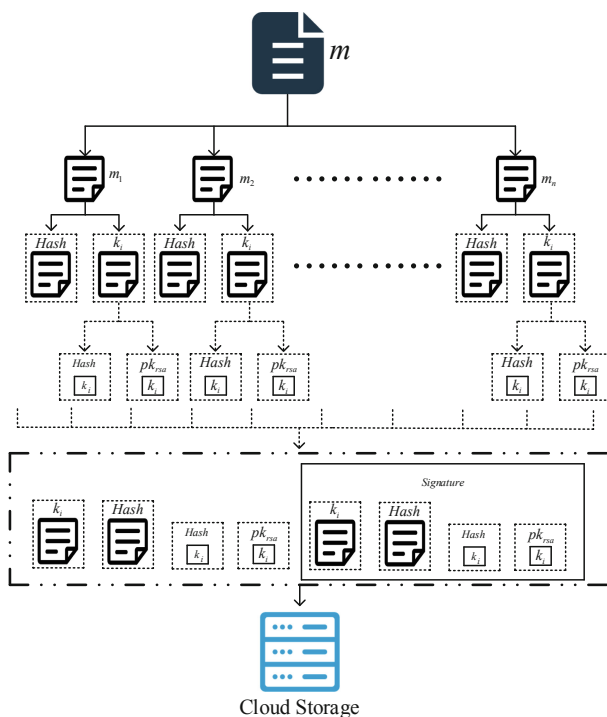


Fig. 3. The framework of data processing.

The whole steps of data processing are described as follows:

1. The DS divides the data m he needs to sell into n equal parts: $\{m_i\}_{i \in \{1, \dots, n\}}$.
2. The DS needs to generate n symmetric key $\{k_i\}_{i \in \{1, \dots, n\}}$, and uses each symmetric key encrypt m_i :

$$c_i = Enc(k_i, m_i) \quad (1)$$

3. The DS use an anti-collision hash function $H(\cdot)$ to compute the hash value of m_i and k_i :

$$h_{m_i} = H(m_i) \quad (2)$$

$$h_{k_i} = H(k_i) \quad (3)$$

4. The DS generates a random number r_i and a pair of RSA public key private key pair (sk_{rsa}, pk_{rsa}) (N is the modulus) and encrypts each symmetric key k_i as follow:

$$s_{1,i} = r_i^{sk_{rsa}} \pmod{N} \quad (4)$$

$$r_{x,i} = H(r_i) \quad (5)$$

$$s_{2,i} = Enc(r_{x,i}, k_i) \quad (6)$$

5. The DS cascades the above documents and sign the cascaded file using the Ethereum wallet address private key:

$$h = \{c_i\}_{i \in 1, \dots, n} \parallel \{h_{m_i}\}_{i \in 1, \dots, n} \parallel \{s_{1,i}\}_{i \in 1, \dots, n} \parallel \{s_{2,i}\}_{i \in 1, \dots, n} \parallel \{h_{k_i}\}_{i \in 1, \dots, n} \quad (7)$$

$$sig_h = sig(sk_{ecc}, h) \quad (8)$$

6. The DS uploads file $h \parallel sig_h$ to the cloud storage, generates the description of the data and record it and the path of the data in the cloud storage on the private blockchain.

4.3 Data Verification

The DP picks up the data he needs from the private blockchain, and views the reputation of the DS from the private blockchain. When the DP finds the right data, downloads the ciphertext and generates a data validation request, and sends it to the DS.

After the DS receives the data verification request from the DP, he checks the times that the DP has verified the data from the private blockchain. If the number of verifications is less than 10 times, the DS records the confirmation information on the private blockchain. Then he verifies the data with the DP.

The whole steps of data verify are described as follows:

- (1) The DS uses a pseudo-random number generator and send it to the DP:

$$x_v = a_{v-1}x_{v-1} + a_{v-2}x_{v-2} + \dots + a_1x_1 + a_0 \quad (10)$$

- (2) The DP randomly generates v numbers, a_0, a_1, \dots, a_{v-1} , and sends them to the DS.
 (3) The DS generates t random numbers, $x_{v+i+1}, x_{v+i+2}, \dots, x_{v+i+t}$. Then he calculates ran and sends it to the DP:

$$ran = Enc_{ecc}(pk'_{ecc}, i || k_{x_{v+i+1}} || k_{x_{v+i+2}} || \dots || k_{x_{v+i+t}} || r_{x_{v+i+1}} || r_{x_{v+i+2}} || \dots || r_{x_{v+i+t}}) \tag{11}$$

(4) The DP decrypts ran and gets the plaintexts:

$$m_{x_j} = Dec(k_{x_j}, c_{x_j})_{j \in \{v+i+1, v+i+2, \dots, v+i+t\}} \tag{12}$$

- (5) The DP checks if the plaintext meets the data description. At the same time, he encrypts the symmetric key again using the pk_{rsa} and r_i to compare whether the symmetric key ciphertext obtained is consistent.
- (6) The DS records the verification record of the DP on the private blockchain.

4.4 Data Purchase

When the user has verified the data, he can initiate a transaction to purchase all symmetric keys of the encrypted data. The data purchaser needs to create a smart contract.

The whole steps of data purchase are described as follows:

- (1) The DS and DP exchange an exchange key g^{xy} by Diffie-Hellman key exchange protocol (where x is the random number selected by DS and y is the random number selected by DP).
- (2) The DS sends $Enc(g^{xy}, Sig(sk_{ecc}, H(g^x) || H(g^y)))$ to the DP, and the DP sends $Enc(g^{xy}, Sig(sk'_{ecc}, H(g^x) || H(g^y)))$ to the DS. The DP and DS decrypt the message from the other party to confirm that the other party has received the correct exchange key g^{xy} .
- (3) The DS generates a random number r , and encrypts r with the shared key, then sends it to the DP.
- (4) The DP creates a smart contract which is shown in the Fig. 4.
- (5) The DP uses the random number r to decrypt the $x : x = sk_{rsa} * r^{-1} \pmod{\varphi(n)}$ by $r * r^{-1} \equiv 1 \pmod{\varphi(n)}$.

<p>Function payment():</p> <ol style="list-style-type: none"> 1) The data purchaser pays \$. 2) The data purchaser sets limitation time T. 3) The data purchaser sets the condition of the transaction, that the data seller can submit x which can satisfy the equation $EM_i^x = s_i^r \pmod{N}$ (k_i and s_i are randomly selected by DP from the verified data).
<p>Function transfer():</p> <ol style="list-style-type: none"> 1) Assert current time $T_1 < T$. <ol style="list-style-type: none"> a) Verify the condition that x submitted by the DS satisfies the equation $EM_i^x = s_i^r \pmod{N}$. b) Send \$ to data seller. 2) Assert current time $T_1 > T$. <ol style="list-style-type: none"> a) Send \$ to data purchaser.

Fig. 4. The smart contract for data trading

- (6) The DP uses sk_{rsa} to decrypt each ciphertext of symmetric key he gets for the cloud storage, and uses the symmetric keys to decrypt the ciphertext of the data:
- (6-1) The DP uses sk_{rsa} to decrypt the ciphertext $s_{1,i} : r_i = s_{1,i}^{sk_{rsa}} \pmod{N}$;
 - (6-2) The DP calculates the hash value of $r_i : r_{x,i} = H(r_i)$;
 - (6-3) The DP calculates the symmetric keys $k_i : k_i = Dec(r_{x,i}, s_{2,i})$;
 - (6-4) The DP recovers out the plaintext: $m_i = Dec(k_i, c_i)$.
- (7) The DP combines the decrypted ciphertexts to get the complete data.
- (8) The DP checks if the plaintext is consistent with his requirements and records the score of the data as the reputation of the DS on the private blockchain.

5 Security and Performance Analysis

5.1 Security Model

This section proposes an IND-CCA security model which is described by an interactive game. We assume \mathcal{A} is an adversary of probabilistic polynomial time (PPT). \mathcal{C} is a challenger. It can be described by the following game between the challenger \mathcal{C} and the adversary \mathcal{A} .

Init: The challenger \mathcal{C} firstly determines two large prime numbers (p, q) and calculates $N = p \cdot q$. Then \mathcal{C} generates a public and private key pair (pk_{rsa}, sk_{rsa}) with (p, q) . Then he keeps the private key sk_{rsacp} as secret value, and sends N and pk_{rsa} to the adversary \mathcal{A} .

Query 1: \mathcal{A} submits a ciphertext (s_1, s_2) to \mathcal{C} , \mathcal{C} runs the decryption algorithm and sends the decrypted message back to \mathcal{A} .

Challenge: \mathcal{A} outputs two messages (K_1, K_2) with same length, and sends them to \mathcal{C} . \mathcal{C} randomly chooses $\beta \leftarrow_R \{0, 1\}$ and calculates $\bar{s}_1 \equiv r^{pk_{rsa}} \pmod{N}$ and $\bar{s}_2 = Enc_r(K_\beta)$. Then \mathcal{C} sends (\bar{s}_1, \bar{s}_2) to \mathcal{A} .

Query 2: The same as query 1, but \mathcal{A} cannot query about (\bar{s}_1, \bar{s}_2) .

Guess: The adversary \mathcal{A} output it's guess β' , if $\beta' = \beta$, we think that the adversary \mathcal{A} is successful.

The advantage of the adversary \mathcal{A} is defined as:

$$Adv^{RSA-CCA}(\mathcal{A}) = |\Pr[\beta' = \beta] - 1/2|$$

5.2 Security Analysis

We show that our scheme can provide IND-CCA security. The security analysis comes from Ref. [21]. For the completeness, we give the detail proof:

Theorem 1 [21]. Based on the above security model, if there exists a PPT adversary \mathcal{A} can solve the RSA problem with the advantage of $\varepsilon(\mathcal{K})$. Then there must be an

adversary \mathcal{B} can attack the IND-CCA security with at least the advantage of $\text{Adv}_{\mathcal{B}}^{\text{RSA}}(\mathcal{K}) \geq 2\varepsilon(\mathcal{K})$.

Proof. We give the definition of the following symbols: in the RSA algorithm, N is the modulus, pk_{rsa} is the public key, sk_{rsa} is the private key, s_1 is part of the ciphertext which means $s_1 \equiv (r)^{pk_{rsa}} \pmod{N}$, r is a random number, r_x is the hash value of r , s_2 is part of the ciphertext which means $s_2 = \text{Enc}(r_x, k)$, Enc/Dec is a symmetric encryption/decryption algorithm and k is a message that need to be encrypted.

We assume that the adversary \mathcal{B} knows (N, pk_{rsa}, \hat{s}_1) , and proceed the following process with \mathcal{A} as a subroutine. The target for the adversary \mathcal{B} is to calculate $\hat{r} \equiv (\hat{s}_1)^{1/pk_{rsa}} \pmod{N}$.

Init. \mathcal{B} selects a random string $\hat{r}_x \leftarrow_R \{0, 1\}^{128}$ as a guess for $H(\hat{r})$, and sends the public key $pk = (N, pk_{rsa})$ to \mathcal{A} .

Oracle \mathcal{O} query 1. \mathcal{B} establishes a \mathcal{O}^{list} , the element type in it is a triple (r, s_1, r_x) , and the initial value is $(*, \hat{s}_1, \hat{r}_x)$, where $*$ indicates that the value of the component is currently unknown. \mathcal{A} can query \mathcal{O}^{list} at any time. If \mathcal{A} queries about r , \mathcal{B} uses r to calculate $s_1 \equiv r^{pk_{rsa}} \pmod{N}$ and responds as follows:

- i. If there exists a triple (r, s_1, r_x) in the \mathcal{O}^{list} , then \mathcal{B} responds with r_x .
- ii. If there exists a triple $(*, s_1, r_x)$ in \mathcal{O}^{list} , \mathcal{B} responds with r_x and replaces $(*, s_1, r_x)$ with (r, s_1, r_x) in \mathcal{O}^{list} .
- iii. Otherwise, \mathcal{B} picks a random number $r_x \leftarrow_R \{0, 1\}^{128}$, responds with r_x and stores (r, s_1, r_x) in \mathcal{O}^{list} .

Decryption query. When \mathcal{A} initiates inquiry $(\overline{s}_1, \overline{s}_2)$ to \mathcal{B} , \mathcal{B} responds as follows:

- i. If there exists a triple in \mathcal{O}^{list} whose second element is \overline{s}_1 (the triple is $(\overline{r}, \overline{s}_1, \overline{r}_x)$), where $\overline{s}_1 \equiv \overline{r}^{pk_{rsa}} \pmod{N}$ or there exists a triple $(*, \overline{s}_1, \overline{r}_x)$, \mathcal{B} responds \mathcal{A} with $\text{Dec}_{\overline{r}_x}(\overline{s}_2)$.
- ii. Otherwise, \mathcal{B} picks a random number $\overline{r}_x \leftarrow_R \{0, 1\}^{128}$ and responds \mathcal{A} with $\text{Dec}_{\overline{r}_x}(\overline{s}_2)$. Then \mathcal{B} stores $(*, \overline{s}_1, \overline{r}_x)$ in \mathcal{O}^{list} .
- iii. **Challenge.** \mathcal{A} outputs two messages (K_1, K_2) with same length. \mathcal{B} randomly chooses $\beta \leftarrow_R \{0, 1\}$ and calculates $\hat{s}_2 = \text{Enc}_{\hat{r}_x}(K_\beta)$. Then \mathcal{B} responds \mathcal{A} with (\hat{s}_1, \hat{s}_2) .

Oracle \mathcal{O} query 2. \mathcal{B} continues to respond the query of \mathcal{O} or decryption from \mathcal{A} , but \mathcal{A} cannot query about (\hat{s}_1, \hat{s}_2) .

Guess. \mathcal{A} outputs his guess β' . \mathcal{B} checks \mathcal{O}^{list} . If there exists $(\hat{r}, \hat{c}_1, \hat{h})$, then he outputs \hat{r} .

In the above view, the values obtained by \mathcal{A} in the oracle query are all random values. And according to the construction of the oracle \mathcal{O} , if \overline{r} corresponding to \overline{r}_x which satisfies $\overline{r}^{pk_{rsa}} \equiv \overline{s}_1 \pmod{N}$ and $\overline{r}_x = H(\overline{r})$, The reply $\text{Dec}_{\overline{r}_x}(\overline{s}_2)$ that \mathcal{B} responds to \mathcal{A} is valid. So the view of \mathcal{A} is indistinguishable from the view in real attacks.

In the above attack, if $H(\hat{r})$ does not appear in \mathcal{O}^{list} , then \mathcal{A} cannot obtain \hat{r}_x . The probability $\Pr[\beta' = \beta | \neg \mathcal{O}] = 1/2$ can be obtained by $\hat{s}_2 = \text{Enc}_{\hat{r}_x}(K_\beta)$ and the IND-

CCA security of *Enc*. It is also known from the definition of \mathcal{A} in the real attack that the advantage of \mathcal{A} is greater than $\varepsilon(\mathcal{K})$, we can get the advantage of \mathcal{A} in the simulated attack is $|\Pr[\text{Exp}_{\mathcal{A}}^{\text{RSA-CCA}}(\mathcal{K}) = 1] - 1/2| \geq \varepsilon(\mathcal{K})$. We can calculate:

$$\begin{aligned}
& \Pr[\text{Exp}_{\mathcal{A}}^{\text{RSA-CCA}}(\mathcal{K}) = 1] \\
&= \Pr[\text{Exp}_{\mathcal{A}}^{\text{RSA-CCA}}(\mathcal{K}) = 1 | \neg\mathcal{O}] \Pr[\neg\mathcal{O}] \\
&+ \Pr[\text{Exp}_{\mathcal{A}}^{\text{RSA-CCA}}(\mathcal{K}) = 1 | \mathcal{O}] \Pr[\mathcal{O}] \\
&\leq \Pr[\text{Exp}_{\mathcal{A}}^{\text{RSA-CCA}}(\mathcal{K}) = 1 | \neg\mathcal{O}] \Pr[\neg\mathcal{O}] + \Pr[\mathcal{O}] \\
&= \frac{1}{2} \Pr[\neg\mathcal{O}] + \Pr[\mathcal{O}] \\
&= \frac{1}{2} (1 - \Pr[\mathcal{O}]) + \Pr[\mathcal{O}] \\
&= \frac{1}{2} + \frac{1}{2} \Pr[\mathcal{O}]
\end{aligned} \tag{13}$$

And we also know that:

$$\begin{aligned}
& \Pr[\text{Exp}_{\mathcal{A}}^{\text{RSA-CCA}}(\mathcal{K}) = 1] \\
&\geq \Pr[\text{Exp}_{\mathcal{A}}^{\text{RSA-CCA}}(\mathcal{K}) = 1 | \neg\mathcal{O}] \Pr[\neg\mathcal{O}] \\
&= \frac{1}{2} (1 - \Pr[\mathcal{O}])
\end{aligned} \tag{14}$$

So we have $\varepsilon(\mathcal{K}) \leq |\Pr[\text{Exp}_{\mathcal{A}}^{\text{RSA-CCA}}(\mathcal{K}) = 1] - 1/2| \leq \frac{1}{2} \Pr[\mathcal{O}]$, which means that in the simulated attack, the probability for \mathcal{B} to win the game is $\Pr[\mathcal{O}] \geq 2\varepsilon(\mathcal{K})$.

In summary, in the above simulation process, \hat{r} appears in $\mathcal{O}^{\text{list}}$ with a probability of at least $2\varepsilon(\mathcal{K})$. And \mathcal{B} will check the elements in $\mathcal{O}^{\text{list}}$ one by one during the guessing phase, so the probability for \mathcal{B} to win the game is equal to the probability of \mathcal{O} .

At present, the RSA problem is still a difficult problem. The advantage of the adversary \mathcal{A} successfully attacking the RSA problem is negligible, so the advantage of the adversary \mathcal{B} successfully attacking our scheme is negligible.

Theorem 2. The proposed blockchain-based data fair trading scheme provides message confidentiality.

Proof: In our scheme, we use AES-128 to instantiate symmetric encryption, and there do not have an efficient algorithm to solve AES-128 so far. Besides, we use RSA algorithm to encrypt the symmetric keys, and when the modulus in the system is sufficiently large, it is impossible to solve the RSA problem. In summary, the probability for the adversary to obtain the useful information of the digital content m is negligible. The message confidentiality is guaranteed.

Anonymity. We use Ethereum's wallet as the user's account information, which itself provides anonymity, and the account information of the private blockchain is only associated with the Ethereum wallet, neither of which reveals the user's identity information.

Ciphertext cannot be falsified. When the Data seller processes the data, we added an anti-collision hash function to calculate the hash value of the segmented plaintext and symmetric key. And the data purchaser can get the plaintext and the corresponding symmetric key in the data verification process. If the hash value submitted by the data seller is incorrect, the data purchaser can easily find it and then refuse to continue the transaction.

Data is undeniable. Here we sign the split data and the hash value of the symmetric key to ensure the source of the data purchased by the data purchaser. There do not have an efficient algorithm to forge a signature signed by ECDSA. And, because we use a reputation system, benign behavior can lead to higher long-term gains. Users who normally trade data will get better long-term gains through honest behavior.

Collusion attack. We mainly analyze the situation where multiple users obtain all data content by verifying data multiple times. Here we assume that the random numbers generated by the random function that we use are close to the true random numbers. And we assume that the data seller divides the original data into 20,50,80 and 100 files, and the data purchaser can verify one of them each time. By changing the number of data verification times, we determine the probability that the data purchaser will get all the data by verifying the data.

As can be seen from the Fig. 5, the less the number that the data seller divides the data into, the easier the data purchaser can get all the data files by multiple verifications. In the Fig. 5(a), we can see that if the data purchaser verifies nearly 70 times (the 3.5 times the current number of data splits) of the same data, he has almost half the possibility to get all the data. In the Fig. 5(b), the number is 220 (the 4.4 times the current number of data splits). In the Fig. 5(c), the number is nearly 380 (the 4.75 times the current number of data splits), And in the Fig. 5(d), the number is nearly 500 (the 5 times the current number of data splits). So we can see that as the data seller increases the number of files that he divides the data more than 100 files, if the data purchasers want to obtain all the plaintext by verification with a relatively large probability, they need to do at least 5 times verification of the number of the divided files.

However, since our verification process is an interactive process, and the data seller can see the identity of the data purchaser, if the single user has maliciously verified multiple times, the data seller can refuse the verification.

Data seller fraud. Here we mainly analyze the situation in which the data seller mixes invalid data in his own data. Here we assume that the data seller divides the data into 100 files, and at the same time, the data seller mixes t useless data in the original data.

In the Fig. 6, we can see the probability that the data seller will fraud successfully when the proportion of the verification data is different. We can find out if the data seller mixes an invalid data in the original data, even let the data purchaser verify 50 times of the data, the data seller still has 50% probability to fraud successfully. But if the data seller does like this, he will only get another $1/99$ profit, he still has 50% probability to fail. And if he fails, this will greatly reduce the possibility for data purchasers buying his data. As data seller increases the proportion of invalid data, the probability for the data seller will dramatic decline. If the data seller mixes 10 percent

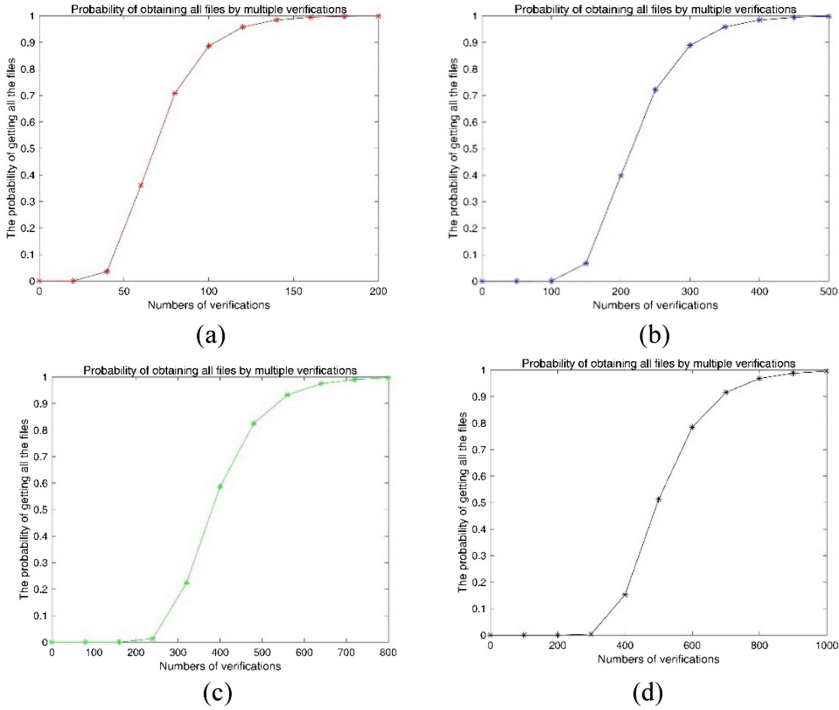


Fig. 5. The probability to obtain all files by multiple verifications.

of invalid data in the original data, if the data purchasers verify the 20 times of the data, the data seller only has 10 percent to fraud.

5.3 Performance Analysis

Compared to other schemes, the server load for all the schemes is low. Only Refs. [4, 7] and our scheme are truly distributed, this ensures that the single point of failure will not happen in the schemes (Table 1).

The scheme in Ref. [3] verifies the data through a trusted party, Refs. [4, 5, 7] and our scheme verifies the data in p2p, and all other schemes do not provide data verification method. For data trading, data authenticity must be guaranteed, thus we must design a suitable method for the data purchasers to verify data. If they verify data through trusted third parties, it is inevitable to leak some information of the data to the trusted third parties, and this may damage to the interests of the data seller.

The schemes in Refs. [1, 2] use the similar method for charge through a bank, which is slightly complicated. References [3, 5] do not provide the method for charge, and Refs. [4, 6, 7] and our scheme provide a simple method for charge. Charge is an indispensable part in data trading, if we trade through the bank, it is a little complicated for data seller and data purchaser to trade the encryption key. The methods in Refs. [4, 6, 7] is similar, but they cannot guarantee the correctness of the encryption key. We use

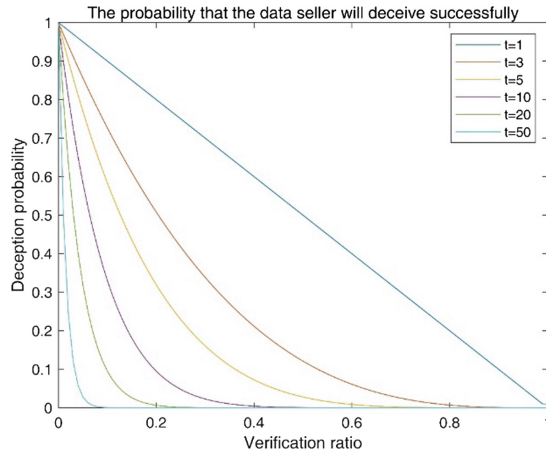


Fig. 6. The probability that the data seller will deceive successfully.

Table 1. Performance comparison

	Server load	Distributed scheme	Verify data	Complexity of the charge	Anonymity	User reputation
[1]	low	×	×	hard	×	×
[2]	low	×	×	hard	√	×
[3]	low	semi	Through Trusted Part	N/A	×	×
[4]	low	√	p2p	easy	√	×
[5]	low	semi	p2p	N/A	√	×
[6]	low	√	×	easy	√	×
[7]	low	semi	p2p	easy	×	×
Our	low	√	p2p	easy	√	√

smart contract to verify the correctness of the encryption key before the data seller get the currency, and use a random number to keep the encryption key as a secret value. It is almost impossible for the data seller to submit a fake encryption key to get the currency.

Only Refs. [2, 4-6] and our scheme can provide the anonymity for the user. Because data trading can turn owned data into wealth, it is necessary to protect user privacy. This will also encourage the people with data to sell data for profit.

Our scheme provides an extra performance by using a private blockchain as a ledger for the user reputation. The data purchaser can decide whether to purchase the data based on the user’s reputation. This will not only reduce the possibility for the data purchasers being deceived, but also let the data sellers maintain a good behavior for better long -term gains.

5.4 Efficiency Analysis

In this section, we analyze the efficiency of the proposed scheme. We contrast the schemes in Refs. [1–7] to our scheme. Here we first assume that all schemes divide the file into 100 files and just pick 3 of them to verify. Here we assume that t_1 is the time of symmetric encryption and decryption, t_2 is the time of the hash function, t_3 is the time of public key encryption, t_4 is the time of private key decryption, and t_5 is the basic time of the smart contract in Ethereum, t_6 is the time of bitcoin transactions, t_7 is the time of one communication, and t_8 is the time of the DAPS algorithm in scheme [7].

As can be seen from Table 2, in the data processing phase, our scheme has significantly more overhead than the other schemes. This is because the scheme in Ref. [1] only uses the symmetric key to encrypt the data which is not convenience for encryption key trading. References [2, 3, 6] do not split the data. Reference [4] uses the public key to encrypt the split files directly. References [5, 7] are similar as our scheme, but they do not use enough hash functions to ensure the data will not be modified and they do not use signature to ensure the source of the data.

Table 2. Efficiency comparison.

	Data processing	Verify data	Data purchase
[1]	$100t_1 + 100t_2$	N/A	$5t_1 + 3t_3 + 4t_4 + 4t_7$
[2]	$10t_2 + 7t_3 + t_4$	N/A	$10t_2 + 12t_3 + 2t_4 + 4t_7$
[3]	$t_1 + 6t_2 + 2t_3 + 2t_4 + 3t_7$	N/A	$9t_2 + 14t_3 + 5t_4$
[4]	$100t_3$	$3t_7$	t_6
[5]	$10t_2 + 100t_3 + 100t_4$	$3t_1 + 3t_7$	$100t_4 + t_7$
[6]	$t_1 + t_2 + 2t_4 + t_7$	N/A	$6t_1 + 4t_2 + 3t_3 + 3t_4 + t_6 + 2t_7$
[7]	$100t_1 + 100t_3$	$3t_1 + 3t_7$	$t_5 + t_8$
Ours	$100t_1 + 400t_2 + 100t_3 + t_4$	$3t_4 + 3t_7$	$t_3 + t_5$

In the data verification phase, Ref. [4] sends the selected plaintext directly for data purchaser to verify. Reference [7] and our scheme send the symmetric key which is used to encrypt the data for data purchaser to decrypt the ciphertext of the data. But other 4 schemes did not provide the method of data verification.

In the data purchasing phase, Ref. [1] uses a bank and the encrypted e-pay to ensure that if the data purchaser has paid the currency. And after the data seller decrypts the encrypted e-pay, the data purchaser will get the symmetric key by some steps. Reference [2] uses a method similar as Ref. [1]. Reference [3] is mainly to exchange the documents, they calculate authorization information and hash function values to exchange the encryption key of the document. References [4, 6] uses the bitcoin transaction script to trade the encryption key. In Ref. [5], data seller and data purchaser directly trade the encryption key after they reach an agreement of the price. Reference [7] and our scheme use the Ethereum smart contract to trade the encryption key, but the calculation method in the smart contract is different. Through our scheme, we can greatly avoid DS submitting fake encryption keys to get money.

Compared to other schemes, our scheme only adds some operations during the data processing phase. Since the data processing of the same data only needs to be done once and is completed before the data trading. The impact on the efficiency of data trading is very small. Moreover, the hash function and signature we added during the data processing stage will improve the security of data trading. In summary, our scheme only sacrifices negligible efficiency, but it increases the security of data trading.

6 Conclusion

In order to solve the problem of security and fairness in data trading, some proposals have been made, such as trading data through trusted third parties or using arbitration to guarantee the fairness and security during the data trading. However, these proposals are vulnerable to single point of failure and may leak useful information of the data. We have designed a fair and verifiable data trading scheme that allows data purchasers to obtain a portion of the data in plaintext for verification without revealing too much information about the data. And once the data seller get the currency, the data purchaser will get the encryption key immediately. More importantly, we maintain the user's reputation information through the private blockchain, so that our scheme can be developed in a benign way. Security analysis shows that our scheme can provide high confidentiality for data, ensure user anonymity, resist collusion attack and data seller fraud. In addition, our scheme is compared with the Refs. [1–7], our scheme provides richer functions to meet the various needs of users in the data trading process. At the same time, our scheme only adds some operations during the data processing phase, at the expense of a small portion of the efficiency, in exchange for higher security.

Acknowledgement. This work is supported in part by the National Key Research and Development Program of China (No. 2016YFB0800601), the Natural Science Foundation of China (No. 61303217, 61502372).

Appendix

1. Oblivious Transfer. Oblivious Transfer is a basic cryptographic primitive that is widely used in areas such as secure multiparty computing.

Oblivious Transfer was first proposed by Rabin [18] in 1981. In his OT protocol, sender S sends a message m to receiver R , and receiver R accepts information m with a probability of $1/2$. So at the end of the interaction, S does not know if R accepted the message.

In 1985 Even, Goldreich, and Lempel proposed 1-out-2 OT [19]. In the new scheme sender S send two messages m_0 and m_1 to R , and R selects a number b as the input. When the agreement ends, S cannot get any useful information from b . R only get the message m_b and cannot get any information of m_{1-b} .

Using the idea of 1-out-2 OT, it is extended to m-out-n OT, allowing the receiver to select m random numbers at a time and accept the data corresponding to the random number of all data sent by the sender. In this way, it is possible to ensure that the data

received by the receiver each time is different, and by reasonably controlling the sizes of m and n , it can ensure that the receiver cannot obtain all of the data within a certain time even if the data is received multiple times.

2. Blockchain. In 2008, the concept of blockchain was first proposed by Satoshi [20]. In the following years, it became a core component of electronic currency bitcoin: as a public ledger for all transactions. The blockchain database can be managed autonomously by leveraging a peer-to-peer network and a distributed timestamp server.

The original blockchain is a decentralized database, which contains a list of blocks that have a growing and well-aligned record. Each block contains a timestamp and a link to the previous block: the design of blockchain makes the data untamperable—once recorded, the data in one block is irreversible.

The blockchain has several important features: 1. Decentralization. Due to the use of distributed accounting and storage, the system does not have centralized hardware or management, and the rights and obligations of any node are equal. 2. Openness. The system is open, the blockchain data is open to everyone, and anyone can query the blockchain data through a public interface. 3. Autonomy. The blockchain uses consensus-based norms and protocols, making trust in “people” a trust in the machine. 4. Information cannot be modified. Utilizing the characteristics of the anti-collision hash function, once the information is verified and added to the blockchain, it is stored permanently, so the data stability and reliability of the blockchain is extremely high. 5. Anonymity. Since the exchanges between nodes follow a fixed algorithm, their data interaction does not require confirmation of the user’s true identity.

3. An application scenario. With the advancement of society, the use of smart devices is getting higher and higher, and the data generated by smart devices will be more and more. At present, many large companies have recommended specific information and services to users through data mining to enhance the user experience. For example, Taobao recommends relevant products to users based on their purchase records and search records. Meituan recommends high-quality catering to users based on their location and evaluation information.

For larger companies, they may only need their own software-generated data for effective data analysis. But for small companies or individuals who are just starting out, they need to purchase data for analysis and research.

Because the data generated by smart devices is often owned by individuals and has limited value. It would be unrealistic to ask them to rely on the sale of these data to make a living. However, selling through a central organization requires worrying about the disclosure of information, failure of the central organization or malicious sale of data, and they are more willing to sell the data they own by one-to-one.

However, at present, such kind of distributed scheme is difficult to achieve data security and fairness of data transactions.

For the data seller Alice who has generated some digital content by smart devices, wants to sell some digital content. On the one hand, she hopes to get some income by selling the digital content, on the other hand, he does not want to pay a fee to a third party. She can sell her digital content through our system.

1. Alice processes data as described in step in Sect. 4.2. She uploads the description, the ciphertext storage path, hash value of the segmented digital content and symmetric keys and the signature on the private blockchain.
2. Assume that Bob wants to get some data. He can search the description on the blockchain. If the description meets his requirement, he can upload his request on the blockchain.
3. After Alice see the request, she works with Bob for data validation as described in step in Sect. 4.3. Through the process of data validation, Bob can see a portion of the plaintext, and can re-encrypt to verify whether Alice is cheating.

If Bob decide to purchase the digital content, he generates a smart contract like Fig. 4. All the information in the smart contract can be obtained through interaction between Bob and Alice. Then he uploads the smart contract on the Ethereum. If Alice can complete the equation in the smart contract, she will get the currency. At the same time, Bob will get the information about the private key. Then he can get the digital content he needs by simply calculating.

References

1. Juang, W.S., Shue, Y.Y.: A secure and privacy protection digital goods trading scheme in cloud computing. In: 2010 International Computer Symposium (ICS 2010), pp. 288–293. IEEE (2010)
2. Chen, C.L., Liao, J.J.: Fair offline digital content transaction system. *IET Inf. Secur.* **6**(3), 123–130 (2012)
3. Hwang, R.J., Lai, C.H.: Provable fair document exchange protocol with transaction privacy for e-commerce. *Symmetry* **7**(2), 464–487 (2015)
4. Delgado-Segura, S., et al.: A fair protocol for data trading based on Bitcoin transactions. *Future Gener. Comput. Syst.* (2017). <https://doi.org/10.1016/j.future.2017.08.021>
5. Kiyomoto, S., Fukushima, K.: Fair-trading protocol for anonymised datasets requirements and solution. In: 2018 4th International Conference on Information Management (ICIM), pp. 13–16. IEEE (2018)
6. Wang, D., Gao, J., Yu, H., Li, X.: A novel digital rights management in P2P networks based on Bitcoin system. In: Li, F., Takagi, T., Xu, C., Zhang, X. (eds.) *FCS 2018*. CCIS, vol. 879, pp. 227–240. Springer, Singapore (2018). https://doi.org/10.1007/978-981-13-3095-7_18
7. Zhao, Y., Yu, Y., Li, Y.: Machine learning based privacy-preserving fair data trading in big data marke. *Inf. Sci.* **478**, 449–460 (2019)
8. Missier, P., Bajoudah, S., Capossele, A., et al.: Mind My Value: a decentralized infrastructure for fair and trusted IoT data trading. In: *Proceedings of the Seventh International Conference on the Internet of Things*, p. 15. ACM (2017)
9. Alrawahi, A.S., Lee, K., Lotfi, A.: Trading of cloud of things resources. In: *Proceedings of the Second International Conference on Internet of things and Cloud Computing*, p. 163. ACM (2017)
10. Perera, C.: Sensing as a service (S2aaS): Buying and selling IoT data. arXiv preprint [arXiv:1702.02380](https://arxiv.org/abs/1702.02380) (2017)
11. Huang, Z., Su, X., Zhang, Y., et al.: A decentralized solution for IoT data trusted exchange based-on blockchain. In: 2017 3rd IEEE International Conference on Computer and Communications (ICCC), pp. 1180–1184. IEEE (2017)

12. Juang, W.S., Shue, Y.Y.: A secure and privacy protection digital goods trading scheme in cloud computing. In: 2010 International Computer Symposium (ICS 2010), pp. 288–293. IEEE (2010)
13. Lin, S.J., Liu, D.C.: An incentive-based electronic payment scheme for digital content transactions over the Internet. *J. Netw. Comput. Appl.* **32**(3), 589–598 (2009)
14. Lin, S.-J., Liu, D.-C.: A fair-exchange and customer-anonymity electronic commerce protocol for digital content transactions. In: Janowski, T., Mohanty, H. (eds.) ICDCIT 2007. LNCS, vol. 4882, pp. 321–326. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77115-9_34
15. Cattelan, R.G., He, S., Kirovski, D.: Prototyping a novel platform for free-trade of digital content. In: Proceedings of the 12th Brazilian Symposium on Multimedia and the Web, pp. 79–88. ACM (2006)
16. Fan, C.I., Juang, W.S., Chen, M.T.: Efficient fair content exchange in cloud computing. In: 2010 International Computer Symposium (ICS 2010), pp. 294–299. IEEE (2010)
17. Qian, W., Qi, S.: A fair transaction protocol with an offline semi-trusted third party. In: Phillips-Wren, G., Jain, L.C., Nakamatsu, K., Howlett, R.J. (eds.) *Advances in Intelligent Decision Technologies*, pp. 249–257. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14616-9_24
18. Rabin, M.O.: How to exchange secrets by oblivious transfer. Report no[R], TR-81, Harvard Aiken Computation Laboratory (1981)
19. Even, S., Goldreich, O., Lempel, A.: A randomized protocol for signing contracts. *Commun. ACM* **28**(6), 637–647 (1985)
20. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
21. Yang, B.: *Provable Security in Cryptography*. Tsinghua University Press, Beijing (2017). (In Chinese)



Public Audit Scheme of Shared Data Based on Blockchain

Junfeng Tian^{1,2}, Xuan Jing^{1,2(✉)}, and Ruifang Guo^{1,2}

¹ Cyberspace Security and Computer College,
Hebei University, Baoding 071000, China
abidble@gmail.com

² Key Laboratory on High Trusted Information System in Hebei Province,
Baoding 071000, China

Abstract. A cloud platform provides users with shared data storage services. While the cloud protects the privacy of users, it is inevitable that malicious users illegally use shared data. Currently, the audit scheme in which managing users access records by group managers is widely adopted, and this scheme realizes the protection of users' identity privacy and the traceability of users' identities. However, this kind of scheme disregards the hidden danger of group managers and has certain limitations. This paper proposes a public audit scheme of shared data based on the blockchain (BBS). First, by introducing the blockchain technology, this paper realizes the sharing of records information, avoids the hidden security risks of group managers, and simultaneously makes the user identity traceable. Second, this paper constructs a novel audit algorithm to pre-process users' revocation, which adopts a new resignature algorithm to make management more secure and reliable. Then, this paper introduces an outsourcing algorithm to reduce the computational burden of users. Finally, the theoretical analysis and experimental verification show that BBS is secure and efficient.

Keywords: Shared data · Blockchain · Resignature algorithm · Outsourcing algorithm

1 Introduction

Cloud storage is one of the most critical services of cloud computing; it provides users with flexible storage space. Via cloud storage, users can outsource their data into the cloud without concerning the data storage and maintenance. Users, however, do not physically possess these data once they store their data into the cloud. Any failure (e.g., hardware failures, external attacks and carelessness of humans) in the cloud may cause disclosure and loss of users' data. To verify the data integrity, a third-party auditor (TPA) is usually introduced for a public audit because the TPA is more powerful than the computing and communication ability of users [1]. Researchers have proposed public audit schemes that support batch data operations [2] and dynamic data operations [3]. However, most schemes only enable a single data owner to access the data, which has some limitations. User data can not only be stored in the cloud but also be shared by multiple users [4].

Many cloud storage service providers (e.g., iCloud, OneDrive, and Baidu Cloud) currently use cloud data sharing as one of their main services. Data sharing enables a group of users to access data that belongs to this group. However, the signature on the shared data may indicate a specific user in the group or a specific data block in the shared data. Some scholars have proposed many shared data audit schemes to protect the identity privacy and data privacy of group members [4–6].

In 2012, Wang et al. [4] proposed a public audit scheme of shared data using ring signature technology, which enables the identity of group members to be hidden to protect the privacy of group members. However, a linear relationship between the length of the signature and the size of the group exists and causes low efficiency in the generation of signature and data integrity verification. Worku et al. [5] employed random mask technology to hide data and constructed homomorphic tags, which guaranteed the data privacy of users in the public audit stage but prompted complicated calculation of the client side. Shen et al. [6] calculated the signature by specifying an agent, which not only protects the identity privacy and data privacy of group members but also realizes the lightweight calculation of group members. However, there may be illegal access to shared data among group members, the scheme does not consider how to revoke the group members who maliciously modify the data.

To realize the dynamic management of groups and support the cancellation of group members, Wang et al. [7] proposed a public audit scheme of shared data using proxy resigning technology to realize the cancellation of illegal group members. However, the scheme may be compromised by collusion attacks by the withdrawn group members and the cloud service provider. Jiang et al. [8] proposed a scheme to revoke illegal group members by a verifier based on vector commitment, which resisted this attack but was inefficient in calculation. Yuan et al. [9] proposed an efficient user revocation scheme and implemented a shared data audit scheme that supported multi-user revocation using a polynomial authentication tag and proxy tag update technologies. Subsequently, Luo et al. [10] applied the concept of Shamir Secret Sharing to realize the efficient withdrawal of illegal group members. However, these schemes [7–10] did not consider the traceability of group membership.

While protecting the identity privacy and data privacy of group members, resisting and curbing the malicious modification of data by illegal group members are necessary. In the case of disputes caused by inconsistent internal data among group members, tracing illegal group members is necessary to resolve disputes. These two aspects are challenges for a public audit of shared data in cloud storage [11].

Currently, a novel group management model proposed by Yang et al. [12] has been extensively adopted. This mode enables a group manager to register/revoke group members, which can not only protect the privacy of group members but also trace the illegal group members. However, the model has some defects: ① This mode adopts the centralized management mode and disregards the uncontrollable factors in the management of group managers. ② Once the authentication tag generated by the group manager's group key is invalid, data validation cannot be completed. ③ The calculation of the authentication tag is complex and expensive. This paper improves the scheme proposed by Yang et al., and proposes a public audit scheme of shared data based on blockchain (BBSD). The main work description of this paper is described as follows:

- (1) We construct a novel cloud storage auditing model for shared data based on blockchain in this paper. In order to realize credible user revocation, we come up with a novel strategy for records management. This design enables group members to supervise each other and legally access data, solves the dispute caused by inconsistent data, reduces the storage burden of group managers, and makes the management scheme in the group more secure and credible.
- (2) We construct a novel audit algorithm to preprocess users' revocation. When the group manager is revoked for some reasons (such as revealing the privacy information of group members or being maliciously attacked), the algorithm is employed to resign the authentication tag generated by the invalid group key, thus, the cloud storage auditing is still effect.
- (3) In our designed audit algorithm, we can introduce the outsourcing algorithm to reduce the calculation cost and make the audit more efficient.

This paper is described as follows: Sect. 2 introduces the model framework and security objectives. Section 3 covers the prerequisites and definitions. Section 4 introduces the specific BBSD scheme. Section 5 covers the security analysis. Section 6 introduces outsourcing algorithms. Section 7 evaluates the performance. Section 8 introduces the summary.

2 Model Framework and Security Objectives

2.1 Model Framework

The shared data framework is shown in Fig. 1(a). The framework model of the BBSD scheme includes five types of entities: Members (M), Group Manager (GM), Regional Manager (RM), Cloud and TPA.

- (1) Group members (M): A group consists of several group members. Any group member can create and upload data files to the cloud on behalf of the group. Any group member can access and modify the shared data of the group in the cloud.
- (2) Group manager (GM): Each group has a group manager. The GM has more storage and computing power than a single group member. The GM helps group members create the authentication tag and is responsible for managing members within the group.
- (3) Regional manager (RM): The RM distributes keys to group managers and group members and is responsible for managing group managers.
- (4) Cloud: The cloud provides storage services for users to share data and provides a platform for group members to share data.
- (5) TPA: The TPA challenges the cloud on behalf of group members for verifying the integrity of shared data. After receiving this challenge, the cloud returns the evidence of shared data to the TPA. Finally, the TPA verifies the correctness of the evidence to judge the integrity of the shared data.

The blockchain is shown in Fig. 1(b), and the transaction records are created by the group members. In a blockchain network, each node (group member and group manager) validates the transaction record and broadcasts it across the network.

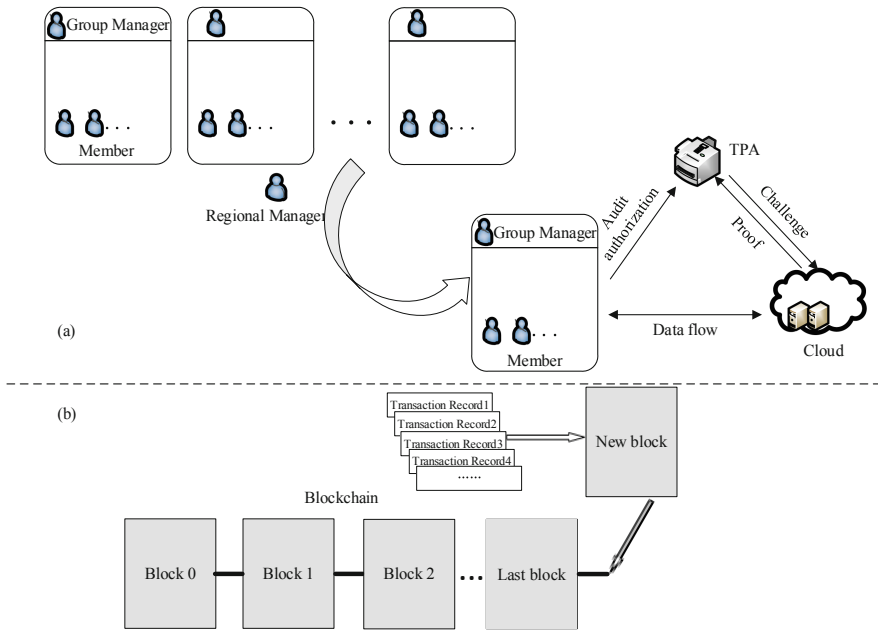


Fig. 1. Frame diagram for data sharing.

After the transaction is verified and received by the nodes in the network, the transaction record is successfully created, and the miner node (refer to Sect. 4.1 for miners’ selection) adds transaction records to a new block on the blockchain that contains many transactions.

2.2 Security Objectives

A well-constructed shared data audit scheme should satisfy the following objectives:

- (1) **Audit correctness:** the cloud accepts the challenge of the TPA, the TPA accepts its proof, and the cloud passes the verification of the TPA. At the end of this process, the cloud should store the shared data with nonnegligible probability.
- (2) **Identity privacy:** during periodic data audits, the TPA cannot obtain the identity information of group members from the authentication tag.
- (3) **Identity traceability:** illegal group members can be identified, and disputes can be resolved using the disputed data.
- (4) **Credibility:** the transaction records are publicly recognized, and the methods for identifying illegal group members and resolving disputes are credible.
- (5) **Data privacy:** when group managers help group members to generate authentication tags, they cannot know the real content of data blocks.
- (6) **High efficiency of management:** the RM can efficiently distribute keys and resignatures to ensure the legal rights of group managers and group members.

3 Preliminary Knowledge and Relevant Definitions

3.1 Preliminary Knowledge

1. Bilinear Maps: Let G_1 and G_2 be two cyclic multiplicative groups with the same prime order p , that is, $|G_1| = |G_2| = p$. Let $e : G_1 \times G_1 \rightarrow G_2$ be a bilinear map, which satisfies the following properties:
 - Bilinearity: $\forall g_1, g_2 \in G_1$ and $a, b \in {}_R Z_p^*$ there is $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
 - Nondegeneracy: For some $g_1, g_2 \in G_1$, $e(g_1, g_2) \neq 1$.
 - Computability: An efficient algorithm exists to compute this map.
2. Computational Diffie-hellman (CDH) Problem: For $x, y \in Z_p^*$, given $g, v = g^x$ and $g^y \in G_1$ as input, output $v^y \in G_1$. The CDH assumption in G_1 holds if solving the CDH problem in G_1 is computationally infeasible.
3. Discrete Logarithm (DL) Problem: For $x \in Z_p^*$, given $g, g^x \in G_1$ as input, output x . The DL assumption in G_1 holds solving the DL problem in G_1 is computationally infeasible.
4. Merkle Hash Tree (MHT): As shown in Fig. 2, MHT is a binary tree consisting of a root node, a set of leaf nodes, and a set of other nodes. The leaf node contains the stored data or its hash value. The root node is the hash of the contents of its two child nodes as well as the other node, such as $h_4 = h(h(n_3) || h(n_4))$. Suppose the verifier has the value h_{root} corresponding to the root node and he wants to verify the integrity of n_4, n_8 . Then, only the certifier needs to provide relevant auxiliary information $\Omega = \{n_4, n_8, h(n_3), h(n_7), h_3, h_5\}$. The verifier could use the auxiliary information to recursively obtain the root node h'_{root} by constructing the MHT and check whether the calculated h'_{root} is the same as the authentic one.

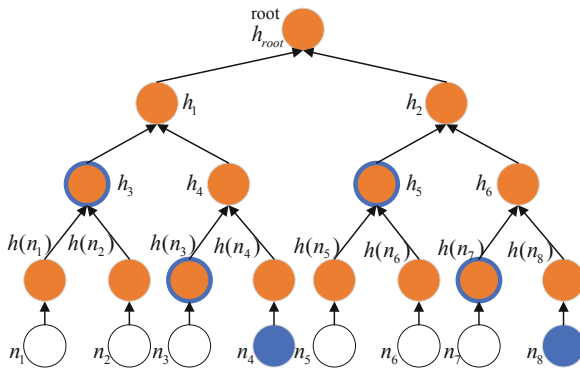


Fig. 2. Merkle Hash Tree.

3.2 Related Definitions

The following definitions are determined according to similar definitions in the literature [13–16]:

Definition 1 (Shared data audit). A public audit scheme of shared data based on blockchain consists of eight algorithms (KeyGen, AuthGen, Resign, Challenge, ProofGen, ProofVerify, EnrollMem, and RevokeMem).

- KeyGen is run by the RM to generate and dispatch system parameters for related entities.
- AuthGen is run by group members to generate authentication tags for shared data.
- Resign is run by the RM. When illegal behavior by the GM is detected, the RM can safely convert the tag to the authentication tag generated by the new group key via the cloud.
- Challenge is run by the TPA to generate challenge information and send it to the cloud.
- ProofGen is run by the cloud, and the integrity proof of shared data is generated according to the received challenge information.
- ProofVerify is run by the TPA to verify the integrity proof provided by the cloud. When validated, the shared data are fully stored in the cloud.
- EnrollMem is run by the GM to add new members to the group.
- RevokeMem is run by the GM to undo group members in a group.

Definition 2 (Blockchain). Blockchain is a kind of data structure that is orderly linked from back to front by blocks, and the blocks in the blockchain contain trading information. As shown in Fig. 3, each transaction record in the blockchain of this paper consists of the identity of the data block id_i , the group membership information of the newly modified block *Sender* and the timestamp of the transaction record t . Each transaction record has the value $h_{root} || \Omega$ corresponding to these contents. New blocks are created from the creation of new transaction records, which are added to the new block by the miners. The block header of each block contains the hash value of its parent block (unique identification of the parent block) [17], and the previous block (parent block) is referenced by its parent block hash value field. The hash value of the block head is linked to a chain that can be traced to the 0th block (creation block).

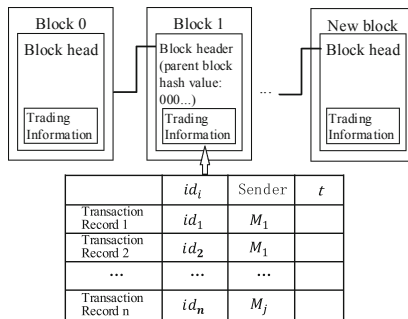


Fig. 3. Blockchain.

4 BBSD Scheme

4.1 Description of Main Knowledge

This scheme proposes a flexible and secure key management method. All keys in the scheme are calculated and distributed by the RM. The group key is a random value $\varepsilon \in Z_p^*$, and the key for each of the group members is a random value $\varepsilon_k \in Z_p^*$ ($k = \{1, \dots, \theta\}$, number of group members is θ). The RM calculates part of the key ε'_k based on $\varepsilon = \varepsilon_k + \varepsilon'_k$. An identity-key List (IKL) is stored by the GM, as shown in Table 1, each record in the IKL is a triple (*No.*, *Sender*, *Key*). When a user registers for the group, the RM randomly generates account M_k as the identity label of the group member. The RM then sends the key ε_k to the group members and part of the key ε'_k to the GM. The GM adds a record to the IKL. When a group member leaves the group, the GM deletes the records of the corresponding group member in the IKL.

Table 1. Identity key list (IKL).

No.	Sender	Key
1	M_1	ε'_1
2	M_2	ε'_2
3	M_3	ε'_3
...
θ	M_θ	ε'_θ

Each group member has access to shared data in the cloud. An operation by which it accesses or modifies shared data as a transaction record that is broadcast between group members and group managers. As shown in Fig. 4 (N represents the entity node), according to the rule of “beating the drum and passing the flower”, miners are selected among group members and group managers. A token is set and passed around the nodes. When a transaction occurs, the token holder is the miner, who puts the transaction record in the new block. A group member can obtain a token when a transaction record is created, at which time the node token is passed to the previous node, which enables the group manager and each group member to have the same opportunity to be the miner and ensures that the group member cannot put the transaction record of this node in the new block when he is a miner.

4.2 Description of BBSD Scheme

The detailed description of BBSD scheme is described as follows:

(1) Algorithm KeyGen(1^k)

1. The RM runs $IG(1^k)$ to generate the cryptographic hash function $H : Z_p^* \rightarrow G_1$, generators $g \in G_1$ and the random value $\alpha \in Z_p$ and then computes $\{g^{\alpha^j}\}_{1 \leq j \leq s}$.

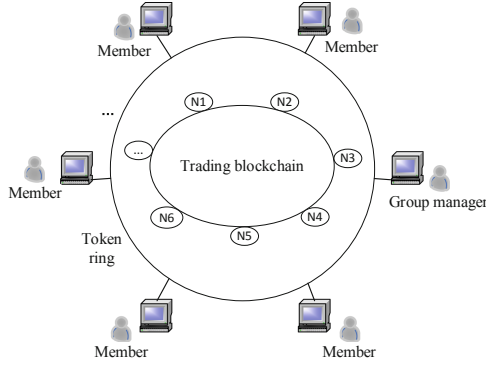


Fig. 4. Token Ring.

The RM send α to the TPA. Note that shared data m is divided into n blocks, and each block has s segments: $\{m_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq s\}$.

2. The RM randomly selects $\varepsilon \in Z_p^*$ as the group key and computes $PK = g^\varepsilon$ as the group public key.
3. The RM randomly selects $\varepsilon_k \in Z_p^*$ ($k = \{1, \dots, \theta\}$) as the secret key of group member M_k and sends ε_k to each group member M_k .
4. The RM computes $\varepsilon'_k = \varepsilon - \varepsilon_k$ as the part of the key and sends them to the GM. The GM initializes the IKL as $\{(M_1, \varepsilon'_1), \dots, (M_\theta, \varepsilon'_\theta)\}$ (as shown in Table 1). The RM sets the global parameter to $\{p, g, n, s, \{g^{\alpha_j}\}_{1 \leq j \leq s}, G_1, G_2, H, PK\}$.

(2) Algorithm AuthGen($m, \varepsilon_k, \varepsilon'_k$)

- (1) M_k randomly selects $r_k \in Z_p^*$ to compute a blind message:

$$m'_i = \prod_{j=1}^s g^{\alpha_j m_{ij}} \cdot g^{r_k} \tag{1}$$

- (2) M_k sends m'_i to the GM and sets (id_i, M_k, t) as the part of $h_{root} \parallel \Omega$ in transaction record to broadcast within the group. Note that (id_i, M_k, t) represents the access records of M_k , and the id_i is the public identifier information for shared data blocks m_i .
- (3) When the GM receives m'_i , it first checks whether M_k is an eligible member. If M_k is not in the IKL, then the GM rejects it. Otherwise, the GM queries the transaction records (id_i, M_k, t) on the block via the blockchain at this time. The GM calculates part of authenticator $\sigma'_i = (H(id_i)m'_i)^{\varepsilon'_k}$ according to m'_i , and id_i corresponds to M_k in IKL; then, the GM sends σ'_i to the member M_k .
- (4) M_k calculates the final authenticator:

$$\begin{aligned}
\sigma_i &= \sigma'_i \cdot (PK/g^{\varepsilon_k})^{-r_k} \cdot (H(id_i)) \cdot \prod_{j=1}^s (g^{\alpha^j})^{m_{ij}} \varepsilon_k \\
&= (H(id_i)) \prod_{j=1}^s (g^{\alpha^j})^{m_{ij}} \cdot g^{r_k} \varepsilon'_k \cdot (g^{\varepsilon - \varepsilon_k})^{-r_k} \cdot (H(id_i)) \cdot \prod_{j=1}^s (g^{\alpha^j})^{m_{ij}} \varepsilon_k \\
&= (H(id_i)) \prod_{j=1}^s (g^{\alpha^j})^{m_{ij}} \varepsilon_k + \varepsilon'_k \cdot (g^{\varepsilon'_k})^{-r_k} \cdot g^{r_k \varepsilon'_k} \\
&= (H(id_i)) \prod_{j=1}^s (g^{\alpha^j})^{m_{ij}} \varepsilon
\end{aligned}$$

M_k sends the data blocks and σ_i to the cloud.

- (5) The cloud verifies the correctness of the authenticator:

$$e(\sigma_i, g) = e(H(id_i) \prod_{j=1}^s g^{\alpha^j m_{ij}}, PK) \quad (2)$$

If the equation holds, the cloud accepts the authenticator from the group member. Group members randomly download the modified authenticator and verify the correctness of the authenticator according to Eq. (2) to ensure the correct implementation of the verification process in the cloud.

(3) Algorithm Resign

- (1) When the group manager GM_1 is revoked from the group, the RM redesignates GM_2 as the new group manager. The RM randomly selects $\varepsilon' \in Z_p^*$ as the new group key. The RM then computes ε'/ε and sends it to the cloud.
- (2) The RM randomly selects $\varepsilon'_k \in Z_p^*$ ($k = \{1, \dots, \theta\}$) and sends it to each group member M_k as its new secret key. The RM regenerates global parameters $\{p, g, n, s, \{g^{\alpha^j}\}_{1 \leq j \leq s}, G_1, G_2, H, PK\}$ and computes part of the key $\varepsilon''_k = \varepsilon' - \varepsilon'_k$. The RM sends ε''_k to GM_2 , and then the GM_2 updates $IKL\{(M_1, \varepsilon'_1), \dots, (M_\theta, \varepsilon''_\theta)\}$.
- (3) The cloud recalculates the authenticator based on the received ε'/ε :

$$\sigma''_i = \sigma_i^{\varepsilon'/\varepsilon} = ((H(id_k) \prod_{j=1}^s (g^{\alpha^j})^{m_{ij}} \varepsilon)^{\varepsilon'/\varepsilon}) = (H(id_k) \prod_{j=1}^s (g^{\alpha^j})^{m_{ij}} \varepsilon')$$

This signature is the valid signature of the group key that corresponds to GM_2 . Therefore, the algorithm Resign is implemented.

(4) Algorithm Challenge

1. The TPA randomly selects c block out of all blocks of shared data and represents the index of the selected block as L .
2. The TPA randomly generates $o, r \in Z_p$ and then computes $X = g^o$ and $R = g^r$.
3. The TPA computes $\{X^{\alpha^j}\}_{1 \leq j \leq s}$.

4. The TPA outputs the challenge messages $CM = \{L, R, \{X^{\alpha_j}\}_{1 \leq j \leq s}\}$ and sends it to the cloud.

(5) Algorithm ProofGen

1. After receiving the challenge message CM , the cloud will generate the proof that the shared data is correctly stored as follows:
2. The index set L of the selected block is divided into the subset L_1, \dots, L_d , where L_i is a subset of the block updated by group member M_i , where $1 \leq i \leq d$.
3. For each subset, the cloud computes $u_{ij} = \sum_{l \in L_i} m_{lj}$ and $\pi_i = \prod_{l \in L_i} e(\sigma_l, R) =$

$$e\left(\prod_{l \in L_i} H(id_l) \prod_{j=1}^s g^{\alpha_j \sum_{l \in L_i} m_{lj}}, g\right)^{e^r}, \text{ where } 1 \leq i \leq d \text{ and } 1 \leq j \leq s.$$

4. The cloud $w_i = \prod_{j=1}^s X^{\alpha_j u_{ij}}$ and $\pi = \prod_{i=1}^d \pi_i$, and then it returns $prf = \{\{w_i\}_{1 \leq i \leq d}, \pi\}$ as a response to the challenge information.

(6) Algorithm ProofVerify

According to prf and CM , the TPA verifies the integrity of shared data by checking the correctness of the following equation:

$$\prod_{i=1}^d e(\eta_i^o, PK^r) \cdot e(w_i, PK^r) \stackrel{?}{=} \pi^o \tag{3}$$

where $\eta_i = \prod_{l \in L_i} H(id_l)$, $1 \leq i \leq d$, and the equation can be further rewritten as

$$\left(\prod_{i=1}^d e(\eta_i^o w_i, PK)\right)^r = \pi^o.$$

If the equation is TRUE, the TPA returns TRUE to the group member. Otherwise, the TPA returns FALSE to the group member. If the selected block in the challenge has been tampered, the cloud will not be able to generate valid evidence and the cloud will not be able to pass the audit process initiated by the TPA.

(7) Algorithm EnrollMem ($M_{\theta+1}, \varepsilon$)

When the user $M_{\theta+1}$ applies to join the group, the RM randomly selects the user's private key $\varepsilon_{\theta+1} \in Z_p^*$ to calculate the partial key $\varepsilon'_{\theta+1} = \varepsilon - \varepsilon_{\theta+1}$, sends $\varepsilon_{\theta+1}$ to the user, and sends $\varepsilon'_{\theta+1}$ to the GM; then, the GM adds $(M_{\theta+1}, \varepsilon'_{\theta+1})$ to the IKL.

(8) Algorithm RevokeMem(M_k)

When a group member leaves the group, the RM notifies the GM to revoke the member and the GM deletes a record in the IKL.

5 Security Analysis

- (1) Correctness: After the cloud returns the evidence prf , the TPA performs the algorithm ProofVerify and verifies the Eq. (3) according to the properties of bilinear mapping. Equation (3) can be proved correct by deducing the right side from the left side.

Proof.

$$\begin{aligned}
& \prod_{i=1}^d e(\eta_i^o, PK^r) \cdot e(w_i, PK^r) \\
&= \prod_{i=1}^d e\left(\prod_{l \in L_i} H(id_l)^o, g^{\varepsilon r}\right) \cdot e\left(\prod_{j=1}^s (g^o)^{x^j \sum_{l \in L_i} m_{lj}}, g^{\varepsilon r}\right) \\
&= \prod_{i=1}^d e\left(\prod_{l \in L_i} H(id_l), g\right)^{o\varepsilon r} \cdot e\left(\prod_{j=1}^s g^{x^j \sum_{l \in L_i} m_{lj}}, g\right)^{o\varepsilon r} \\
&= \prod_{i=1}^d e\left(\prod_{l \in L_i} H(id_l)\right) \prod_{j=1}^s g^{x^j \sum_{l \in L_i} m_{lj}}, g^{o\varepsilon r} \\
&= \prod_{i=1}^d \pi_i^o = \left(\prod_{i=1}^d \pi_i\right)^o = \pi^o
\end{aligned}$$

- (2) Identity privacy: In the current scheme, the user's key is usually employed to calculate the authentication tag. Since the key and the user's identity are uniquely bound, the TPA can infer the user's identity information of the corresponding data block [4]. In our scheme, the data block is actually bound with the group key. The TPA cannot know the membership by the group public key, that is, from the perspective of the TPA, the final authentication tag can be generated by anyone in the group, and the probability is equal. If the group has d members, the probability of correctly guessing is $1/d$. Therefore, the scheme can protect the identity privacy of users.
- (3) Identity traceability: Two types of illegal group members exist: ① If a malicious group member modifies data blocks, the dirty data blocks may be identified by other group members. Once the group has produced controversy, group members can search the transaction records to identify all group members, who have access to shared data. The member sequential modification on the data block enables legal group members to open the data block to prevent illegal framing by a group member and eventually open the illegal data block, which is the illegal member. ② If the transaction records in the disputed block differ from those recognized by the group members, the miners are illegal members.

When group members issue the request to generate partial authentication tags, the (id_i, M_k, t) in transaction records of updated data are disclosed, and the group managers

obtain this transaction and other relevant information (m'_i) to help them generate partial authentication tags (σ'_i). All data modification records are disclosed in the transaction records, including malicious transaction records. When group members search for transaction records, they can learn the identity of dishonest members and realize the traceability of group membership.

- (4) Data privacy: Before sending the data block to the group manager, the group members blind the data block according to Eq. (1). Since the DL problem is computationally infeasible, the group managers cannot obtain $\prod_{j=1}^s g^{\alpha^j m_{ij}}$ from m_{ij} . The blind data were further processed by the random value g^{r_k} ; thus, the group manager cannot infer useful information from the combined operation of multiple blind information.
- (5) Credibility: Scheme [12] solves disputes via group managers. Once the method of dispute resolution cannot be convinced by legal group members (group managers illegally address disputes and cover up illegal users), obtaining reasonable and effective evidence to expose malicious group managers or group members will be impossible. In our scheme, the method of blockchain is adopted, which ensures the correctness of transaction records, restrains the illegal behaviors of group members, and solves disputes in a more secure and credible way.
- (6) Efficient resignature: The resigning algorithm improves the efficiency of revoking illegal group managers. The RM can realize resigning without downloading data blocks [6, 18]. This method reduces the communication overhead and the computational complexity. The group key cannot be calculated by the resignature parameter ε'/ε , which ensures the security and realizes the efficient resignature.

6 Outsourcing Algorithm

Exponential and bilinear pairings are expensive computations but are the primary computations in the cloud storage integrity audit scheme [19, 20]. Therefore, secure computing outsourcing is needed to reduce the computing overhead and render it more suitable for mobile computing environments.

As previously mentioned, the algorithm $\text{AuthGen}(m, \varepsilon_k, \varepsilon'_k)$ of the BBSD scheme requires group members to calculate the product of a large number of power operations, which causes calculation burden for group members. To improve the efficiency, the scheme is extended. According to the outsourcing algorithm in scheme [10], this scheme needs to outsource part of the authentication tag σ'_i .

$$\sigma'_i = (H(id_i)m'_i)^{\varepsilon'_k}, \text{ where } m'_i = \prod_{j=1}^s g^{\alpha^j m_{ij}} \cdot g^{r_k}.$$

By the outsourcing algorithm,

$$\sigma'_i = (H(id_i)m'_i)^{\varepsilon'_k}, \text{ where } m'_i = \phi_1 \varphi_1^{\chi_1} \cdot g^{r_k}.$$

ϕ_1 and φ_1 are generated by the cloud providing outsourced computing services. Group members select the random value χ_1 ($\chi_1 \geq 2^\lambda$, where λ is the 160-bit safety parameter) to calculate $\varphi_1^{\chi_1}$ and then calculate σ'_i .

7 Performance Evaluation

7.1 Numerical Analysis

By outsourcing the polynomial authentication tag, the computation cost of the authentication tag generation phase is reduced.

For the convenience of analysis, the following symbols are used to represent the specific calculation cost: Mul_{G_1} , Mul_{G_2} and Mul_{Z_p} represent the multiplication times in G_1 , G_2 , and Z_p , respectively. Exp_{G_1} and Exp_{G_2} represent the exponential operation times in G_1 and G_2 , respectively. Sub_{Z_p} represent the subtraction time in Z_p . $Pair$ is the time to compute a bilinear pair $e : G_1 \times G_1 \rightarrow G_2$, and $Hash$ is the time to compute the hash function.

Table 2 shows the comparison of calculation cost between this scheme and scheme [12]’s authentication tag generation algorithm.

Table 2. Authentication Tag Generation Calculation Cost Table.

Calculation overhead	Scheme [12]	BBSD
Blind data	$n(2Exp_{G_1} + Mul_{G_1})$	
Blind data (outsourcing)		$2sMul_{Z_p} + 2(d - 1)s \cdot Sub_{Z_p} + 2Exp_{G_1} + 2nMul_{G_1}$
Partial authentication tag	$n(Exp_{G_1} + Mul_{G_1} + Hash)$	$n(Exp_{G_1} + Mul_{G_1} + Hash)$
Final authentication tag	$n(2Exp_{G_1} + 5Mul_{G_1} + Hash)$	$n(2Exp_{G_1} + 5Mul_{G_1} + Hash)$
Summation	$n(5Exp_{G_1} + 7Mul_{G_1} + 2Hash)$	$n(3Exp_{G_1} + 8Mul_{G_1} + 2Hash) + 2sMul_{Z_p} + 2(d - 1)s \cdot Sub_{Z_p} + 2Exp_{G_1}$

Table 3 shows the comparison of the communication costs in the audit stage between this scheme and scheme [12]:

Table 3. Communication Overhead during the Audit Phase.

Communication overhead	Challenge stage	Proof stage
Scheme [12]	$cs(p + id)$	$d(G_1 + p)$
BBSD	$c n + (s + 1) G_1 $	$d G_1 + G_2 $

Note: $|n|$ is the size of the element, $|p|$ is the size of the element in Z_p^* , $|id|$ is the size of the block identifier, $|G_1|$ is the size of the element in G_1 , and $|G_2|$ is the size of the element in G_2 .

7.2 Experimental Results

Let ω be the number of damaged shared data blocks, n is the number of shared data blocks, ρ is the number of damaged data blocks selected by TPA in the audit process, and P_ρ is the probability that TPA selects a damaged shared data block. We can calculate

$$P(\rho = 0) = 1 - P_\rho = \frac{n-\omega}{n} \cdot \frac{n-1-\omega}{n-1} \dots \frac{n-c+1-\omega}{n-c+1},$$

where $1 - (\frac{n-\omega}{n})^c \leq P_\rho \leq 1 - (\frac{n-c+1-\omega}{n-c+1})^c$.

According to the analysis of the literature [10, 12], the TPA can detect damaged data blocks in the shared data with a very high probability. In addition, the probability of detecting corrupt data blocks is independent of the total number of data blocks. For example, if 1% of the shared data blocks are corrupted, the TPA can detect the corrupted data block with 95% probability by checking 300 blocks. Similarly, if the TPA checks 460 blocks, it can detect corrupted data blocks with 99% probability.

The following experiments were conducted to evaluate the performance of the scheme. Pairing-Based Cryptography (PBC) library functions are used to simulate Cryptography operations. The size of the element in Z_p^* is $|p| = 160$ bits. The block identifier size is $|id| = 20$ bits. The shared data size is 2 MB, which consists of 10000 blocks, and $s = 100$. The experiment is implemented in the Ubuntu operating system. The computer’s processor is Inter Core i7 3.4 GHz, and the memory is 4 GB. The experimental results were the average of 10 experiments.

(1) Calculation cost of authentication tag generation.

By calculation outsourcing, the experimental results shown in Fig. 5 show that the time cost of authentication tag generation linearly increases with an increase in the number n of shared data blocks (note: when $n = 0$, the authentication tag generation process is not executed, and $2sMul_{Z_p} + 2(d - 1)s \cdot Sub_{Z_p} + 2Exp_{G_1}$ is negligible when n is large.). According to Fig. 5, the computational cost of the scheme in this paper is lower than that of the authentication tag generation in the scheme [12]. Therefore, our audit scheme reduces the calculation cost of the users and makes the audit more efficient.

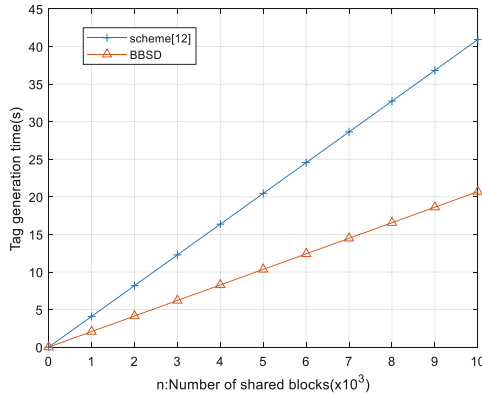


Fig. 5. Authentication tag generation overhead.

(2) Verification costs in the audit phase.

Due to the powerful computing power of the cloud, more attention is often paid to the overhead of the client in the process of a data integrity audit. To further analyze the cloud audit cost, the verification cost of this scheme was tested. By analyzing the audit process of this scheme, we know that the verification cost of the BBSD scheme in the audit stage is $dExp_{G_1} + 3dMul_{G_1} + (2d + 1)Exp_{G_2} + 2dMul_{G_2}$. As shown in Fig. 6, the verification time linearly increases with the number of group members. When the number of group members is 200, the verification time is approximately 59 ms.

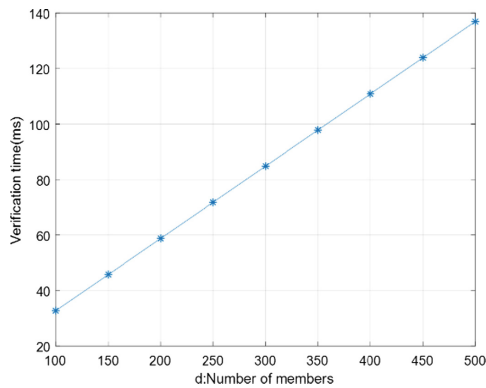


Fig. 6. Verification overhead for a different number of group members.

(3) Communication overhead in the audit phase.

According to the previous analysis, when the number of challenged data blocks is $c = 460$, the data integrity can be guaranteed. These blocks are employed to evaluate the audit communication overhead of the BBSD scheme. As shown in Fig. 7, the communication cost in the audit stage of this scheme is less than that in scheme [12]. During the audit phase of this scheme, the communication overhead linearly increases with the number of group members. When the number of group members increases to 100, the overhead of this scheme consumes approximately 16 KB of bandwidth. This overhead is more efficient for the TPA and group members.

(4) Storage overhead of group managers.

The IKL is stored by the group manager, and the storage overhead is shown in Fig. 8. Each tuple in the IKL contains group membership information, partial keys, and timestamps, for a total storage cost of approximately 30 bits. As the total number of records increases from 1 to 100,000, the maximum storage cost is 2.5 MB, which indicates that group managers do not require a large storage overhead.

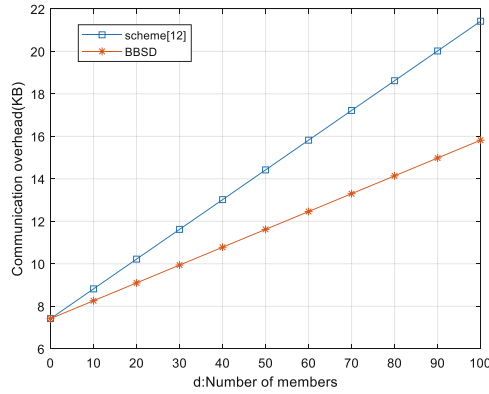


Fig. 7. Communication overhead for a different number of group members.

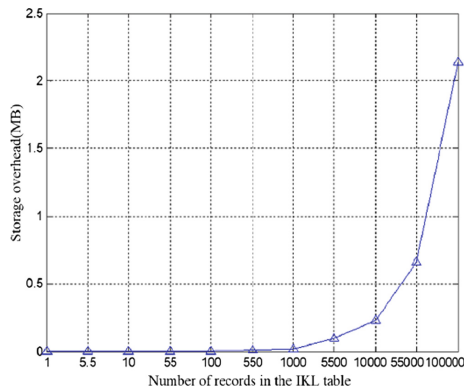


Fig. 8. IKL table storage overhead.

7.3 Robustness

The RM is usually a trusted user (the data owner) who handles complaints filed by group members and group managers.

To ensure that the group manager and other group members are aware of the data modification record and to ensure that the group members cannot independently generate valid authentication tags independently, our scheme utilizes BLS signature technology [19] to build an interactive authentication tag generation process between group managers and group members and via blockchain technology to public transaction records.

Because the CDH problem is computationally infeasible, group members cannot generate valid partial authentication tags without partial keys nor can they forge the final authentication tags. When group members or external users randomly guess the element in G_1 as the authentication tag, the probability of success is $1/p$, where p is the order of G_1 . Because p is very large, the possibility of group members or external users who independently generate valid authentication tags is negligible.

The cloud server verifies $e(\sigma_i, g) \stackrel{?}{=} e(H(id_i) \prod_{j=1}^S g^{2^j m_{ij}}, PK)$. If the equation is true, the final authentication tag is correct and the group manager is legally involved in the interactive authentication tag generation process.

The group manager stores the IKL to ensure that some of the authentication tags are correctly generated. If the group manager does not register/cancel the group members according to the IKL, the group members can provide feedback to the RM. The RM shall periodically review the correctness of the IKL and receive complaints from group members. If any problem is identified in the management of the group manager, the RM shall revoke the group manager.

The identity information of group members is disclosed within the group. Even if all transaction records are obtained by group members, the actual identity of each group member cannot be determined according to this account. In the case of a data dispute, legitimate group members can appeal to the regional manager according to their accounts and then find the group members in the actual scenario.

8 Conclusion

In this paper, we propose a public audit scheme of shared data based on the blockchain (BBSDB). Our scheme protects the privacy of group membership and realizes the traceability of user identity in a decentralized way. We introduce a new management technology blockchain to make the management mode credible. We design a new audit algorithm, which can efficiently undo group managers and improve the management form. And we introduce an outsourcing algorithm to our scheme to reduce the computational burden. Finally, a comparison of the better public audit scheme of shared data indicates that the management mode of this scheme is more secure and efficient.

References

1. Hao, Y., Li, J.G., Han, J.G., Zhang, Y.C.: A novel efficient remote data possession checking protocol in cloud storage. *IEEE Trans. Inf. Forensics Secur.* **12**, 78–88 (2016)
2. Li, Y., Yao, G., Lei, L.N., Wang, H.Q., Lin, C.L.: Large branching tree based dynamic provable data possession scheme. *J. Inf. Sci. Eng.* **33**, 653–673 (2017)
3. Wang, F., Xu, L., Wang, H.Q., Chen, Z.D.: Identity-based non-repudiable dynamic provable data possession in cloud storage. *Comput. Electr. Eng.* **69**, 521–533 (2018)
4. Wang, B.Y., Li, B.C., Li, H.: Oruta: privacy-preserving public auditing for shared data in the cloud. *IEEE Trans. Cloud Comput.* **2**, 43–56 (2014)
5. Worku, S.G., Xu, C.X., Zhao, J.N., He, X.H.: Secure and efficient privacy-preserving public auditing scheme for cloud storage. *Comput. Electr. Eng.* **40**, 1703–1713 (2014)
6. Shen, W.T., Yu, J., Xia, H., Zhang, H.L., Lu, X.Q., Hao, R.: Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium. *J. Netw. Comput. Appl.* **82**, 56–64 (2017)
7. Wang, B.Y., Li, B.C., Li, H.: Public auditing for shared data with efficient user revocation in the cloud. *IEEE Trans. Serv. Comput.* **8**, 92–106 (2015)

8. Jiang, T., Chen, X.F., Ma, J.F.: Public integrity auditing for shared dynamic cloud data with group user revocation. *EEE Trans. Comput.* **65**, 2363–2373 (2016)
9. Yuan, J.W., Yu, S.C.: Efficient public integrity checking for cloud data sharing with multi-user modification. In: *Proceedings of the 2014 IEEE Conference on Computer Communications*, Toronto, Canada 27 April 2014–2 May 2014
10. Luo, Y.C., Xu, M., Huang, K., Wang, D.S., Fu, S.J.: Efficient auditing for shared data in the cloud with secure user revocation and computations outsourcing. *Comput. Secur.* **73**, 492–506 (2018)
11. Fu, A.M., Qin, N.Y., Song, J.Y., Su, M.: Privacy-preserving public auditing for multiple managers shared data in the cloud. *J. Comput. Res. Dev.* **52**, 2353–2362 (2015)
12. Yang, G.Y., Yu, J., Shen, W.T., Su, Q.Q., Fu, Z.J., Hao, R.: Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability. *J. Syst. Softw.* **113**, 130–139 (2016)
13. Ateniese, G., et al.: Provable data possession at untrusted stores. In: *Proceedings of the 14th ACM Conference on Computer and Communications Security*, Alexandria, VA, USA, 29 October 2007–11 December 2007
14. Erway, C., K p c , A., Papamanthou, C., Tamassia, R.: Dynamic provable data possession. In: *Proceedings of the ACM Transactions on Information and System Security (TISSEC)*, Chicago, IL, USA, 9 November 2009–13 November 2009
15. Juels, A., Kaliski Jr., B.S.: PORs: proofs of retrievability for large files. In: *Proceedings of the 14th ACM Conference on Computer and Communications Security*, Alexandria, VA, USA, 29 October 2007–2 November 2007
16. Shacham, H., Waters, B.: Compact proofs of retrievability. In: Pieprzyk, J. (ed.) *ASIACRYPT 2008*. LNCS, vol. 5350, pp. 90–107. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89255-7_7
17. Mastering Bitcoin. Homepage. http://book.8btc.com/books/1/master_bitcoin/_book/. Accessed 13 Dec 2018
18. Zhang, X.P., Xu, C.X., Zhang, X.J., Gu, T.Z., Geng, Z., Liu, G.P.: Efficient dynamic integrity verification for big data supporting users revocability. *Information* **7**, 31 (2016)
19. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. *J. Cryptol.* **17**, 297–319 (2004)
20. Shen, W.T., Yu, J., Yang, G.Y., Cheng, X.G., Hao, R.: Cloud storage integrity checking scheme with private key recovery capability. *J. Softw.* **27**, 1451–1462 (2016)

System and Network Security



A Hybrid Key Management Scheme for Wireless Sensor Network

Yanyan Han^{1,2} , Yanru He¹  , Peihe Liu¹, Xiaoxuan Yan¹,
and Na Li²

¹ Beijing Electronic Science and Technology Institute,
7 Fufeng Road, Beijing, China
hyr63476984@163.com

² Xidian University, 2 Taibai South Road, Xi'an, Shaanxi, China

Abstract. With the rapid development of the Internet of things (IoT), the wireless sensor network (WSN) as the most fundamental layer of the network is widely applied to the IoT, and more researchers focus on the security of WSN. Intrusion Detection System (IDS) is an important element in the security of computer system and smart devices. In this paper, a key management scheme which designed for WSN and based on area management is proposed under the premise that the system has IDS, and the scheme divides network into a number of non-overlapping hexagonal areas. In our scheme, two different key management modes are used for inter-regional and intra-regional communication respectively, and the certificates and keys of gateway node and cluster-head node can be efficiently managed by introducing the security gateway. The scheme not only can reduce the complexity of computation and storage effectively, but also improve the communication security and network connectivity.

Keywords: WSN · Key management · Identity · Authentication

1 Introduction

In recent years, the Internet of things has been known as the third wave of information era, and the wireless sensor network has broad application prospects in military detection, target tracking, situational awareness and other fields due to their advantages of high redundancy, low power consumption, self-organization and rapid deployment [1]. However, WSN is different from other traditional network because of resource-constrained sensor nodes, all of these make it difficult to run efficiently in the high-strength encryption algorithm, and most of nodes are generally deployed in exposed environment, it is easy to be physically accessed by adversaries, the key information stored in nodes can also be stolen. The adversary attacks the whole network by forging, tampering and so on, then finally leads to the collapse of the whole network. So, the key management is an essence of WSN security, and it is the security basis to manage the key in a reasonable and order way. Therefore, we propose a hybrid key management scheme, which divides WSN into several non-overlapping hexagonal network areas. Meanwhile, the security gateway is introduced to distribute security certificates to cluster-head nodes and gateway node, then it manages both certificates and keys

effectively. Our scheme supports the key revocation and update of nodes, and has the ability to resist a variety of attacks. On the condition of ensuring high key connectivity, it effectively reduces the communication, computing and storage complexity of nodes, it also improves the network security compared with other schemes.

The remainder of the paper is organized as follows: Sect. 2 discusses some of classic key management schemes in WSN. According to the drawbacks of existing schemes, a hybrid key management scheme is proposed based on the characteristics of clustering wireless sensor network in Sect. 3. Section 4 gives the performance analysis of the proposed scheme. Finally, we conclude the superiority of the proposed scheme.

2 Related Work

Wireless sensor network nodes are generally deployed in extreme conditions, so most of them has the limitation of power, computing capability and storage capacity. Therefore, due to the characteristics of WSN, traditional key management schemes cannot apply, but key pre-distribution is an effective solution. Key pre-distribution scheme refers to the distribution of keys at the time of node deployment, and it only needs simple negotiation between nodes to encrypt the sessions during the network is running [1, 2]. Now numerous key management schemes based on key pre-distribution are proposed for WSN.

The earliest random key pre-distribution scheme is given by Eschenauer and Gligor [3] as basic scheme where a key pool is established first and each sensor node then randomly selects one of the keys in the key pool as the keyring for that node. Assuming that a communication link is to be established between two neighbor nodes, they must share the same key of the key ring, and a part of key is randomly selected as the key pair in communication links. This scheme has the advantages of low computational complexity, low storage pressure of nodes, and strong adaptability in dynamic network. But network connectivity is not high, and the key is selected based on probability, so the security is low. Once the node is captured by adversaries, it will threaten to the security of the link. Subsequently, Chan et al. [4] proposed the q-composite scheme. Compared with E-G [3] scheme, it requires that the nodes share at least q keys to establish a secure communication link. This scheme enhances the network security by improving the q value, but the network connectivity is worse. At the same time, the security of WSN will decrease rapidly with the increasing number of captured nodes. In [5–7], a series of improvement schemes and methods are put forward to solve such problems as low key connectivity, limited scale of network, high communication and storage cost. However, any compromised node will directly affect the security of key information in the entire WSN, and the key connectivity has not been effectively improved in these key management mechanisms. In [8], Blom et al. proposed a matrix-based key management mechanism. This mechanism not only allows any two nodes to establish a secure connection but also effectively improves network connectivity. In terms of security, the mechanism can ensure the network absolute safety unless more than λ nodes are compromised. Then an efficient key establishment and update mechanism based on Blom scheme is given by Hussain et al. [9]. But the scheme fails to solve the problem of threshold value λ , which means the security of the key is still

restricted to the threshold value λ . Deployment based key pre-distribution is presented by Du et al [10]. The design of the scheme is to implement a simple security connection and improve the network connectivity, simultaneously reduce the storage and computing requirements of nodes. But it cannot meet the expansion of the network, and the location information cannot be accurately obtained due to node position errors. Blundo et al. [11] gives a scheme based on polynomials, using the symmetry of polynomials to generate session keys between nodes. In this scheme, the communication cost of key establishment is reduced, and the storage cost of sensor nodes is reduced to ensure the expansion capacity of network to some extent. When the compromised nodes are less than t , the network is absolutely safe. However, with the increase of t , the storage overhead and computational complexity of nodes also increase sharply, which will shorten the network life to a certain extent, even result in a poor distributed management in the network. The LEAP scheme is given by Zhu et al. [12] which provides a beautiful idea for the application of key management for dynamically clustering, it can support multiple communication mode of WSN and a strong anti-destroying ability. Besides, any compromised nodes will not affect the others, and the scheme also provides authentication function, can resist the wormhole attack and so on. But the master key of the whole network will be saved by all nodes. Once the master key compromised, the network will be crashed. In addition, the scheme cannot support the nodes added and cannot adapt to the dynamic of network. Group key management schemes based on logical routing tree is proposed in [13] and [14]. The ultimate purpose of these schemes is to complete the establishment of group key to ensure multicast communication security, and reduce the cost in WSN. The existing key management schemes are unable to meet all requirements related to security, storage, computation and communication of WSN. Thus, a key management scheme is needed to improve the network connectivity, key management efficiency and communication security on the premise of ensuring low computing and storage costs.

3 The Proposed Scheme

According to the shortcomings of existing solutions, this paper proposes a hybrid key management scheme based on the assumption that the system has intrusion detection function, combined with the characteristics of clustering wireless sensor network. And this section details the working of the proposed scheme. The symbols and their meanings are listed in Table 1.

Intrusion Detection System (IDS) is an important element in the security of computer systems and smart devices, it can detect malicious actions and respond. There are various forms of responses, and the most common of which is to create an alert announcing an enemy invasion. However, intrusion detection system is not responsible for resisting intrusion [15], and its main functions are as follows: monitor device or user behavior, find and respond to suspicious activities, and report them to the administrator.

In contrast to common sensor nodes, the cluster-head node has stronger computing power, more sufficient energy and storage. Therefore, our scheme is based on the structure of clustering type and adopts the area management model divided network

Table 1. Symbols and their meanings

Symbol	Meaning
C_i	Regular hexagon grid area
K	Key space
m_{ij}	Node
$ID_{m_{ij}}$	Identifier of node
K_m	Space of shared key
$K_{m_{ij}m_{ik}}$	Session key
L_i	List of compromise node
Mes_{UK}	Key update message
h_i	Cluster-head node
GWN	Update message sent by the gateway node
Cer	Security certificate
CA	Certificate Authority
rand	Random number generated by requester and responder
rand*	The random number decrypted by using private key of requester
rand**	The random number decrypted by using private key of responder

into non-overlapping hexagonal areas. The model is given in Fig. 1, and particularly, the non-overlapping hexagonal area is the leak-free area coverage model with the least repetition, which can guarantee the high connectivity of the network, and the specific proof can be known in [16]. Each area has a cluster-head node and several common sensor nodes. At the same time, the scheme includes both inter-area and intra-area key management schemes. The node in the area generates the core key by using the Blom matrix, and the session key of Inter-area is established by cluster-head node. The security gateway distributes security certificates to the cluster-head node and gateway node, and takes responsible for the certificate and key management between them.

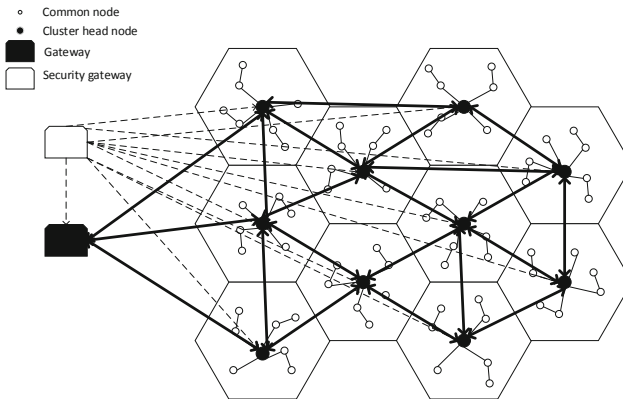


Fig. 1. The clustering type WSN architecture based on area management

In this structure, each hexagonal network area has a cluster-head node and the same number of common nodes. The security gateway as the root CA, distributes and manages the certificates for the cluster-head nodes and gateway node.

3.1 The Inter-area Key Management Scheme

Compared with the common sensor nodes, the cluster-head node owns more computation and communication capabilities, as well as more sufficient energy and storage. Thus, the cluster-head node is selected in our scheme as the communication bridge, and the session key between areas is established by the cluster-head node in each area. Security gateway is introduced as root CA in the scheme, and used for certificating and managing gateway node and cluster-head nodes. Nevertheless, if gateway communicate with cluster-head nodes, the security gateway does not participate. The security gateway is mainly composed of five parts, including main control board, security management module, key agreement module, identity authentication module and encryption and decryption module. The main control module is the control center which manages key and certificate of both the cluster-head and gateway nodes, the key agreement module and the identity authentication module are used to the key agreement and identity authentication between nodes, and the secret key is generated in the encryption and decryption module. According to whether the gateway node is connected to the IP network and send data to the cloud platform, the working mode is divided into online or offline mode. In the online mode, the gateway node manages the key and certificate of the nodes in WSN. While the gateway node and cluster-head nodes are managed by the security gateway with certificate and key management in the offline mode. Figure 2 shows the diagram of security gateway, and the structure of cluster-head node is shown in Fig. 3. Where the arrows in the two figures represent the relationships and data flow interactions between the important components of the nodes.

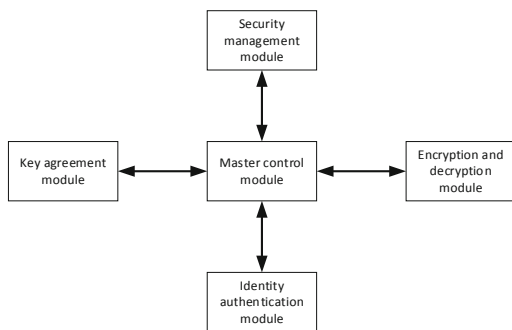


Fig. 2. The structure of security gateway node

In the offline mode, the security gateway selects gateway node and cluster-head nodes to establish the safety management network. The security gateway updates the certificates and the root CA public key of gateway node and cluster-head nodes for authentication. The session key is required in the step of key agreement before updating,

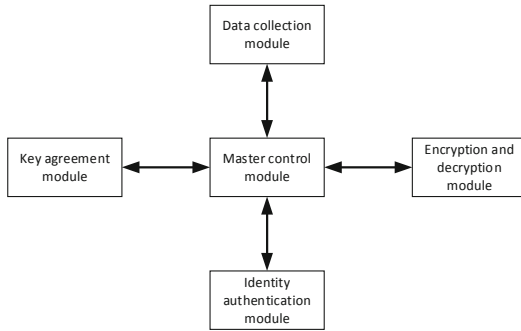


Fig. 3. The structure of cluster-head node

and the certificates of gateway and all cluster-head nodes must be updated after get the new public key. The process of inter-area security communication is as follows:

- Step.1 Gateway and cluster-head nodes start identity authentication and key agreement to obtain session key;
- Step.2 The gateway gets the cluster-head nodes information by ciphertext transmission;
- Step.3 The cluster-head nodes in each area transmit data with ciphertext.

Safety binding is required for cluster-head nodes before deploying the network in each area. The flowchart of cluster-head nodes bound by security gateway is shown in Fig. 4.

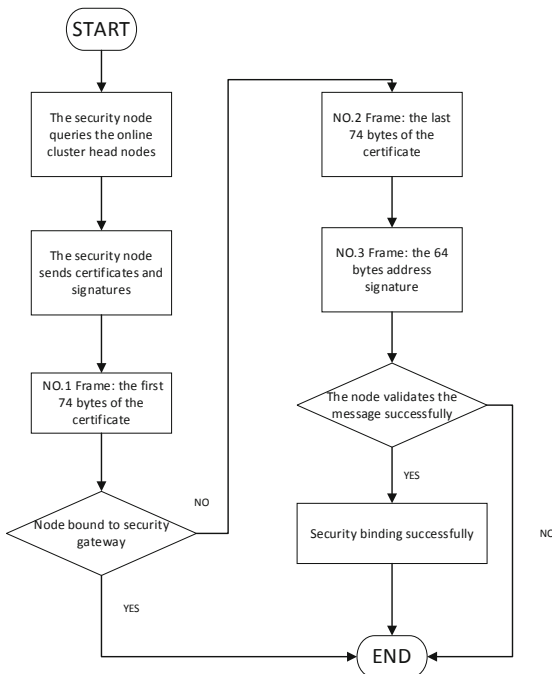


Fig. 4. The cluster-head node safety bound by the security gateway

In the scheme, the length of the data for security binding commands is 212 bytes. One of the 148 bytes are the security gateway certificate or gateway certificate and the rest of 64 bytes are the signature. The security gateway transmits the data as three frames. The first frame is the first 74 bytes of the certificate, containing the information whether the node is bound or not. After receiving the first frame, the cluster-head node will respond a secure bound message to the security gateway. The data of second frame is the remaining 74 bytes of the certificate. In the third frame, the 32-bytes address of the security gateway that sent the binding command is signed as a 64 bytes value. Then the cluster-head node will respond the data frame by frame, which is the secure binding command sent by the security gateway. After the first frame is sent, the remaining two frames are stopped immediately. Then the cluster-head node validates the first frame to verify whether the node has been bound or not. If the binding is done, it is the time to send the address information to security gateway and end the conversation. Otherwise, security gateway will continue to send the second frame, and the cluster-head node will receive and store the second frame data. Then, the security gateway continues to send the address signature information of the third frame, and the cluster-head node will call for the certificate from the security module to assure whether the message is sent by the security gateway. After passing the verification, the address of the cluster-head node is sent to the security gateway to complete the binding. If the binding is failed, it will return a 16-bit-all-zero message. After that, the identity authentication and key agreement between the gateway node and the cluster-head node should be carried out. Both gateway node and cluster-head node can initiate a request for confidential communication. For convenience, the party initiating the communication is called Requester A, and another party is called Responder B. The communication interaction process is shown in Fig. 5.

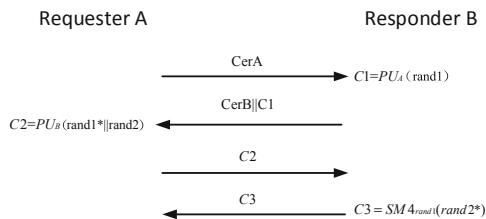


Fig. 5. Authentication and key agreement of the Inter-area

The detailed process of communication is as follows:

- (1) Requester A reads its certificate $CerA$ and sends it to Responder B.
- (2) Responder B reads the root public key, then uses the key to verify the certificate $CerA$ sent by Requester A, and gets the public key PU_A of Requester A. Then, Responder B generates the random number $rand1$, and the SM2 algorithm is used to encrypt $rand1$ to get $C1$. The encryption key is the public key of the requester PU_A . Finally, the certificate $CerB$ of responder is transmitted along with $C1$ to the Requester A.

- (3) After receiving the message, Requester A uses the root public key to verify the certificate Cer_B sent by responder B and get the public key of responder B PU_B . Then, using its private key to decrypt the received information C1, get $rand1^*$, and generate random number $rand2$. After that, the public key PU_B is used as the encryption key and the SM2 algorithm is used to encrypt $rand1^*$ and $rand2$ to get C2. Finally, C2 is sent to Responder B.
- (4) After receiving the message C2 sent by Requestor A, Responder B decrypts it with its own private key and gets $rand1^{**}$ and $rand2^*$. And then determine whether $rand1^{**}$ and $rand1$ are equal. If not, end the session. Otherwise, use $rand1$ as the encryption key and use SM4 algorithm to encrypt $rand2^*$ to get C3. Finally, C3 is sent to Requestor A.
- (5) Firstly, the Requestor A uses $rand1^*$ as the decryption key to get $rand2^{**}$ after receiving the information C3. Then, if $rand2^*$ and $rand2$ are equal, the authentication is successful and $rand1$ is used as the encryption key. Otherwise, authentication fails and the session ends.

Above all, the authentication and key agreement mode between cluster-head node and security gateway can effectively realize the authentication and ensure the security of identity information during the process of communication.

3.2 The Intra-area Key Management Scheme

In the intra-area key management scheme, cluster-head node and common node are not distinguished. The session keys between the nodes use the Blom matrix to generate the core keys, and this scheme gives a different key space for each area, then allocates the key of each node according to the node's ID and other deployment information.

(1) Key pre-distribution phase

In the initialization stage, the server first constructs a $(t + 1) \times N$ Vandermonde matrix in the finite field $GF(q)$. And t is the security threshold of the shared key, if t or more sensor nodes are compromised in the network, the WSN is not secure. Take Area C_i as an example, there are m nodes in C_i , and each node has a unique ID, where Area C_i represents the Vandermonde matrix of C_i then we can get matrix G_i as follows:

$$G_i = \begin{bmatrix} 1 & 1 & \dots & 1 \\ (ID_{m_{i1}})^1 & (ID_{m_{i2}})^1 & \dots & (ID_{m_{iN}})^1 \\ (ID_{m_{i1}})^2 & (ID_{m_{i2}})^2 & \dots & (ID_{m_{iN}})^2 \\ \vdots & \vdots & \ddots & \vdots \\ (ID_{m_{i1}})^t & (ID_{m_{i2}})^t & \dots & (ID_{m_{iN}})^t \end{bmatrix} \quad (1)$$

Then, the server randomly generates a $(t + 1) \times (t + 1)$ symmetric secret matrix D_i in the finite field $GF(q)$, and the Blom matrix A_i of the area can be obtained:

$$A_i = (D_i \cdot G_i)^T \quad (2)$$

The Key Space is the matrix $K = A \cdot G$, and it is represented by K_i as follows:

$$\begin{aligned}
K_i &= A_i \cdot G_i \\
&= (D_i \cdot G_i^T) \cdot G_i \\
&= G_i^T \cdot D_i \cdot G_i \\
&= (A_i \cdot G_i)^T \\
&= K_i^T
\end{aligned} \tag{3}$$

Then, taking m_{ij} as an example, any node in Area C_i only need to store the $\text{row}_j(A_i)$ of matrix A_i , and the identifier of node $ID_{m_{ij}}$ in advance.

(2) Key establishment phase

The key information stored by all nodes in the regular hexagon grid area comes from the matrix G_i and A_i . Node m_{ij} and node m_{ik} in C_i store information $\{ID_{m_{ij}}, \text{row}_j(A_i)\}$ and $\{ID_{m_{ik}}, \text{row}_k(A_i)\}$ according to the key pre-distribution respectively, the space of shared key between node m_{ij} and m_{ik} is $K_m = (A_m \cdot G_i)$.

Firstly, node m_{ij} and node m_{ik} exchange their node identification information ID with each other. Then the node m_{ij} is computed by using $ID_{m_{ik}}$:

$$\begin{cases} \text{col}_k = [1 & (ID_{m_{ik}})^1 & (ID_{m_{ik}})^2 & \cdots & (ID_{m_{ik}})^t]^T \\ K_{m_{ij}m_{ik}} = [\text{row}_j(A_i) & \times & [\text{col}_k(G_i)] \end{cases} \tag{4}$$

Then we can get the session key $K_{m_{ij}m_{ik}}$ between node m_{ij} and m_{ik} from symmetry:

$$K_{m_{ij}m_{ik}} = K_{m_{ik}m_{ij}}, \tag{5}$$

(3) The key updates and revocation

When the gateway node detects that a node has been compromised by an adversary in the WSN, the system can revoke the session key established. Intrusion detection as an active defense technology can prevent internal and external attacks [15], which has become a strong security premise for WSN. The premise of our scheme is that the system has intrusion detection function [17, 18], and the system sends the list of compromised nodes to the gateway node for key update. The process of key update and revocation is shown in Fig. 6.

Suppose that the compromise node list is $L_i = (m_{i1}, m_{i2}, \cdots, m_{ik})$ in the regular hexagon grid area, and the steps of update are as follows:

- A. In the finite field, the gateway node randomly generates a new $(t+1) \times (t+1)$ secret matrix, then it can be obtained from the regular hexagon grid area C_i :

$$\begin{cases} A_i^* = (D_i \cdot G_i)^T \\ \text{Sum}_i = A_i + A_i^* \end{cases} \tag{6}$$

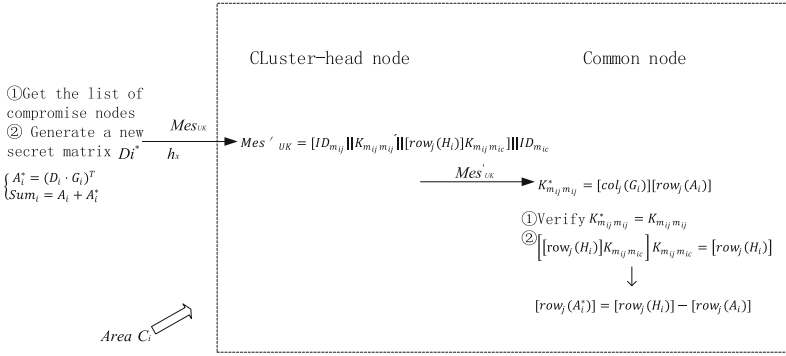


Fig. 6. The key update and revocation process

- B. Set up an n-dimensional row vector R_i . If node m_{ij} is in list L_i , then $R_i[j] = 0$, otherwise, $R_i[j] = 1$.
- C. Create a matrix $H_i = Sum_i \cdot R_i$.
- D. The gateway node sends the update key message to the cluster-head node h_i of C_i . Assuming that the message passes in the middle cluster-head node h_x , then:

$$\begin{cases} K_{m_{ij}m_{ij}} = [col(G_i)] \cdot [row(A_i)] \\ Mes_{UK} = [ID_{m_{ij}} || K_{m_{ij}m_{ij}} || [row_j(H_i)] | 1 \leq j \leq n] K_{h_x, h_i} || GWN \end{cases} \quad (7)$$

And GWN means that the message is sent by the gateway node, and later nodes can use $K_{m_{ij}m_{ij}}$ to verify the correctness of the message. K_{h_i, h_x} represents the encryption of $[ID_{m_{ij}} || K_{m_{ij}m_{ij}} || [row_j(H_i)] | 1 \leq j \leq n]$, after the cluster-head node h_i receives the message, we uses the stored shared key K_{h_i, h_x} to decrypt the message and gets $[ID_{m_{ij}} || K_{m_{ij}m_{ij}} || [row_j(H_i)] | 1 \leq j \leq n]$.

Then, the cluster-head node h_i in C_i sends the key update message Mes_{UK} to all legitimate common nodes based on the node identifier in the area:

$$\begin{aligned} h_i(m_{ic}) &\rightarrow m_{ij}: \\ Mes'_{UK} &= [ID_{m_{ij}} || K'_{m_{ij}m_{ij}} || [row_j(H_i)] K_{m_{ij}m_{ic}}] || ID_{m_{ic}} \end{aligned} \quad (8)$$

When the common node receives the key update message, it will calculate according to its stored information:

$$K^*_{m_{ij}m_{ij}} = [col_j(G_i)] [row_j(A_i)] \quad (9)$$

Then to judge $K^*_{m_{ij}m_{ij}} = K_{m_{ij}m_{ij}}$, if not equal that we can make sure the message is not from the gateway and discard it. Otherwise, the key is updated.

$$\begin{aligned}
 & [[\text{row}_j(H_i)]K_{m_jm_{ic}}]K_{m_jm_{ic}} = [\text{row}_j(H_i)] \\
 & [\text{row}_j(A_i^*)] = [\text{row}_j(H_i)] - [\text{row}_j(A_i)] \tag{10}
 \end{aligned}$$

Finally, we can get new key information $\text{row}_j(A_i^*)$. For compromised nodes, because of the $\text{row}_j(H_i)$ is 0, key updates cannot be performed, and the later data communication and other operations cannot be carried out. By judging the key information, the node can receive the message of the gateway node in time, then update the key and communicate.

3.3 The Application of Proposed Scheme

Our scheme divides the network into hexagonal sub-areas with full coverage, so it mainly applied to the perception layer with cellular architecture in the Internet of Things. The prominent advantage of our scheme is and realize the central key management and authentication through cluster-head nodes in each area, so as to simplify the communication process and reduce the cost. At the same time, the security gateway is introduced to enhance the ability of anti-attacking. The performance analysis and comparison are described specifically in Sect. 4.

The application process of the scheme is shown as follows: Firstly, in the offline mode, the security gateway starts to bind and identify cluster-head nodes and gateway nodes, so as to realize certificate and key management. Secondly, in the online mode, the security gateway stops working. If we do intra-area communication, cluster-head nodes as same as other common nodes start key agreement to obtain communication keys and, then complete secret communications. If we do inter-area communication, the gateway will set up a secure network with each cluster-head node, then obtain the communication key through identification, and complete the secret communication. Especially, the security gateway usually works in the offline mode. Once the gateway node is attacked, the security gateway will start the emergency response and work in the online mode temporarily instead of the gateway. Moreover, the introduction of security gateway enhances the system’s anti-attacking with the acceptable overhead.

4 Performance Analysis of the Proposed Scheme

4.1 Security

4.1.1 Security of Compromised Node

In the scheme, if anyone wants to attack a key space successfully, he must gain $t + 1$ nodes in that area. If the total number of nodes does not exceed $t + 1$, the network is absolutely safe. In addition, when the gateway node finds that the sensor node is compromised, it will send the update message to the node in that area. After other common nodes receive the message of key update, it obtains the new key information through calculation. However, the operation of key update cannot be completed for the compromised node. Therefore, our scheme shows the probability of key leakage through any compromised node is 0 in the hexagon area.

Inter-area communication is conducted by cluster-head nodes. We introduce a security gateway as the root CA to accomplish certificate and key management for gateway nodes and cluster-head nodes. Even if the gateway node and the cluster-head node are controlled by the attacker, the important information such as the communication key and the pairing key will not be disclosed. At the same time, bidirectional authentication performed firstly in communication, and a new session key is created each time which can effectively prevent malicious nodes, and establish trust between the nodes. Therefore, even if the previous session key is captured, the attacker cannot get other session keys. In inter-area communication, the scheme chooses the three-way authentication based on certificate, so it can resist replay attack. Figure 7 shows the ratio of the number of compromised nodes to the number of links affected.

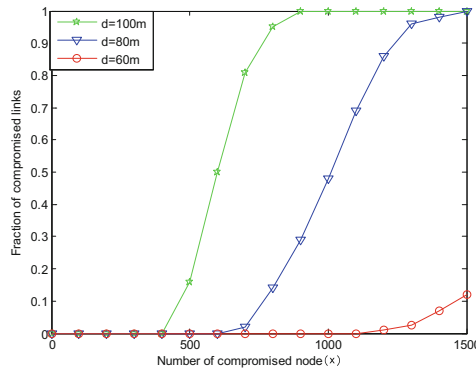


Fig. 7. The ratio of the number of compromised nodes to the number of links affected

Assuming that the number of compromised nodes is x , and the probability of the key space information contained in each node is expressed as $p = \mu/\omega$, Then the expression of probability that the number of compromised nodes and the corresponding key space are all broken by the attacker is as follows:

$$P = \sum_{i=t+1}^x C_x^i p^i (1-p)^{x-i} \tag{11}$$

Among them, d represents the side length of the regular hexagon area. It can be concluded that with the expansion of area and the same number of compromised nodes, the proportion of affected communication links will increase accordingly. That is, the more compromised nodes are concentrated in an area, the more vulnerable the area is to an attacker.

4.1.2 Random Attack Security Analysis

Random attack means that the adversary attacks the network without knowing the node distribution and key management scheme. For a better comparison with the existing solutions, the deployment densities of the nodes should be the same. Figure 8 shows the comparison of the scheme’s ability to resist random attacks with E-G [3] scheme

and q-composite [4] scheme. It can be concluded from the experimental results that our scheme has better anti-random attack ability, and when the number of nodes acquired by the adversary is equal, the influence of the communication link suffered takes up the smallest proportion.

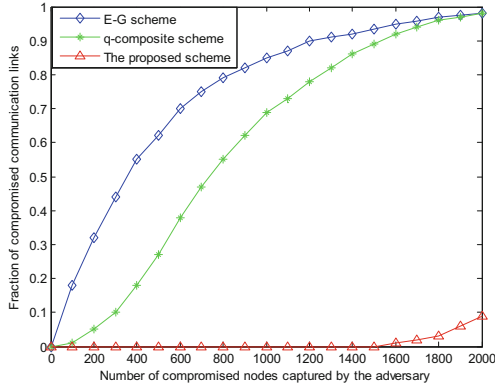


Fig. 8. Comparison of anti-random attack schemes

(1) Resist physical capture attacks

In the inter-area communication, the scheme introduces a security gateway as the root CA which is the key distribution center, and manage the certificate and key for the gateway node and cluster-head node. Even if those critical nodes, like the gateway node and the cluster-head nodes, are captured and controlled by the adversary, the group communication key, broadcast key, pairing key and other important information will not be revealed. When communicating inside the area, take area C_i as an example, the node m_{ij} only needs to store the $row_j(A_i)$ and $ID_{m_{ij}}$. Even if the node is captured and controlled by the attacker, the group key and other important information will not be leaked. Therefore, our scheme can resist physical capture attack no matter it is inter-area or intra-area communication.

(2) Resist eavesdropping attack

In each authentication process of inter-area communication, the cluster-head node and gateway node will generate a new session key randomly, so the new generated key is different from the previously generated. Even if the previous session key is captured, the adversary cannot get other session keys. In the process of intra-area communication authentication, when the gateway node detects that a compromised node, the system can revoke the established session key. But all of above are under the premise of the system has intrusion detection function [17, 18], and the system sends the list of compromised nodes to the gateway node for key update. In conclusion, the scheme can prevent eavesdropping effectively.

(3) Resist replay attack

The authentication process of inter-area communication adopts the three-way authentication based on digital certificate. Since the random number is generated randomly, the third party cannot know it, so when verifying the signature sent by the other party, it can determine whether the message has been modified in the session by comparing the generated random number is equal to itself. After the gateway node of the intra-area finds that the sensor node is compromised, it will send the update message to other nodes in the area. After other nodes receive the key update message, it updates the new key by calculating. Therefore, the adversary cannot conduct a replay attack to the WSN through the compromised node and the scheme can resist replay attack.

(4) Resist fake node attacks

In the proposed scheme, both sides of any communication are bidirectional authenticated before the session key is generated in the inter-region communication. The asymmetric key system is used between the gateway node and the cluster-head node. Both sides of the communication use the public key for encryption, and the private key is used for decryption, which has strong security. For the cluster-type network, key management adopted in this scheme, it is most vulnerable to the cluster-head fake attack of the adversary, and the adversary obtains the node identifier of the cluster-head node after compromise the common node, and then fake as the cluster-head node to broadcast the information to other common nodes. This scheme has the function of intrusion detection, which can detect any compromised node, and once it detected, the it will update the key immediately. When the cluster-head node communicates with the common node, it will first verify whether the key update message is from the gateway node, and if so, the key update operation will be completed. Otherwise, the session ends. In the inter-area communication, if there is a compromised node, the gateway will take the action of key update, and the compromised node cannot complete an update and obtain the communication key. Therefore, the scheme can resist node fake attack.

4.2 Connectivity Analysis

For the inter-area communication established by cluster-head nodes, a security gateway is introduced as the root CA to manage the certificate and key of both gateway node and cluster-head nodes. Before the cluster-head nodes communicate, bidirectional authentication and key agreement should be complete. Therefore, secure sessions can be established between any cluster-heads node, and the connectivity is 1. For the nodes in the intra-area, any node can establish the session key by exchanging the node identifier ID with the neighboring node, so as to achieve the secure communication. Therefore, a secure session can be established between any node in the area for so the connectivity is also 1.

Therefore, the network connectivity of our scheme is always 1. The comparison figure of the network connectivity between the proposed scheme and E-G [3] scheme and q-composite [4] scheme is shown in Fig. 9.

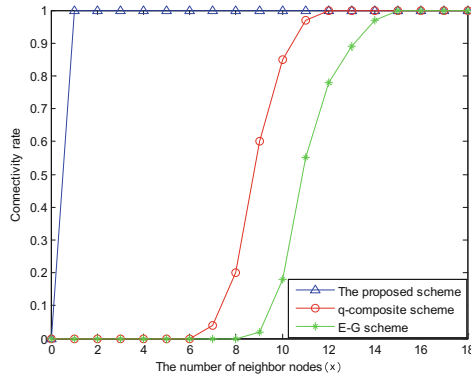


Fig. 9. Comparison of the network connectivity

As it can be seen from Fig. 9, in our scheme, as long as the number of neighbor nodes is not zero, the network connectivity rate is 100%. For E-G [3] scheme, the network connectivity rate can reach 100% only when the number reaches 14 or more. While the q-composite [4] scheme has the worse one, and the number must reach 16 or more, so that the network connectivity rate can reach 100%.

4.3 Performance Analysis of Storage

Common nodes in the intra-area only need to exchange their identifier IDs with each other, and the session key can be established without key agreement. In addition, common nodes need to store only the column of information corresponding to the public matrix G and the row of information corresponding to the Blom matrix A . The required space of single node to store key information is: $k = (t + 1) * \mu + t + 1$. At the same time, because there is not the same key space between each regular hexagon grid region, it can contribute to isolation, so it avoids a lot of unnecessary communication consumption. The relationship between the number of key spaces and the number of areas required is shown in Table 2.

Table 2. Total required of key space

Total required number of keys	2	3	4	5	6
The number of areas	2	14	20	12	126

The cluster-head node has fewer neighboring nodes, so it can reduce the communication cost of key establishment. At the same time, most of the energy in WSN is used for communication, so the reduction of communication overhead can reduce the lifetime of WSN. The scheme is compared with E-G scheme [3], q-composite scheme [4] and LEAP scheme [12] in terms of computational, communication and storage complexities. The comparison results are shown in Table 3.

Table 3. Performance comparison of key management schemes

	Storage complexity	Computation complexity	Communication complexity
E-G scheme [3]	$O(k)$	$O(k)$	$O(2)$
q-composite scheme [4]	$O(m)$	$O(m)$	$O(2)$
LEAP scheme [12]	$O(d+l)$	$O(d^2/N)$	$O(\log N)$
The proposed scheme	$O(2)$	$O(2)$	$O(2)$

As shown in Table 3, the storage cost, computing cost and communication cost of our scheme are greatly reduced compared with others.

5 Conclusion

WSN as the perception layer is the lowest layer of the standard three-layer architecture of the Internet of things. The WSN nodes are often deployed in extreme conditions with limited resource. Therefore, traditional key management is not suitable for WSN. The hybrid key management scheme proposed in this paper is based on the premise that the system has intrusion detection function. It introduces security gateway to manage the key and certificate of gateway node and cluster-head nodes, so as to effectively prevent the key information leakage. In our scheme, the network is divided into non-overlapping hexagonal network regions with a cluster-head node and a number of common sensor nodes. In the inter-area, the communication overhead is reduced and the security of communication is improved by key agreement and bidirectional authentication between cluster-head nodes. In the intra-area, the key pre-distribution of identifiers is exchanged between nodes, which greatly reduces the computing cost, thus achieving high efficiency of key management. Compared with other classical schemes, the results are shown the higher security and efficiency of key management, as well as the better network connectivity.

References

1. Yu, B., Zhou, W., Bin, Y., et al.: ZigBee model for detection and suppression of same-frequency attacks. *J. Electron. Inf. Technol.* **37**(9), 2211–2217 (2015)
2. Lofallahtabrizi, P., Morgan, Y.: A novel host intrusion detection system using neural network. In: *Computing and Communication Workshop and Conference*, pp. 124–130. IEEE (2018)
3. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: *ACM Conference on Computer and Communications Security*, pp. 41–47 (2002)
4. Chan, H., Perrig, A., Song, D.: Random key predistribution schemes for sensor networks. In: *Proceedings of 2003 Symposium on Security and Privacy*, pp. 197–213. IEEE (2003)

5. Du, W., Deng, J., Han, Y.S., et al.: A key management scheme for wireless sensor networks using deployment knowledge. In: IEEE INFOCOM 2004: Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies, Hongkong, pp. 586–597. IEEE (2004)
6. Huang, D., Mehta, M., van de Liefvoort, A., et al.: Modeling pairwise key establishment for random key predistribution in large-scale sensor networks. *IEEE/ACM Trans. Netw.* **15**(5), 1204–1215 (2007)
7. Wang, H., Yang, J., Wang, P., Tu, P.: Efficient pairwise key establishment scheme based on random pre-distribution keys in WSN. In: Taniar, D., Gervasi, O., Murgante, B., Pardede, E., Apduhan, Bernady O. (eds.) ICCSA 2010. LNCS, vol. 6018, pp. 291–304. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12179-1_26
8. Blom, R.: An optimal class of symmetric key generation systems. In: Beth, T., Cot, N., Ingemarsson, I. (eds.) EUROCRYPT 1984. LNCS, vol. 209, pp. 335–338. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39757-4_22
9. Hussain, A.W., Ibrahim, M.K.: An efficient pairwise and group key management scheme for wireless sensor network. *Int. J. Enhanc. Res. Sci. Technol. Eng.* **1**(4), 25–31 (2015)
10. Du, W., Deng, J., Han, Y.S., et al.: A pairwise key predistribution scheme for wireless sensor networks. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **8**(2), 228–258 (2005)
11. Blundo, C., Santis, A.D., Herzberg, A.: Perfectly-secure key distribution for dynamic conferences. *Inf. Computat.* **146**(1), 1–23 (1998)
12. Zhu, S., Setia, S., Jajodia, S.: LEAP: efficient security mechanisms for large-scale distributed sensor networks. In: ACM Conference on Computer and Communications Security, Dallas, USA, pp. 62–72. ACM (2003)
13. Jiang, R., Luo, J., Wang, X.: HRKT: a hierarchical route key tree based group key management for wireless sensor networks. *KSII Trans. Internet Inf. Syst.* **7**(7), 2042–2060 (2013)
14. Jiang, R., Luo, J., Wang, X.: A logic-route key tree based group key management scheme for wireless sensor networks. In: 2013 IEEE/CIC International Conference on Communications in China (ICCC), Xi'an, pp. 686–691. IEEE Computer Society (2013)
15. Du, Y., Zhang, Y., Li, M., et al.: Optimization method of intrusion detection sample data based on improved FastICA algorithm. *J. Commun.* **37**(1), 42–48 (2016)
16. Zhao, S., Zhang, Z.: Research on regular hexagon node coverage model of wireless sensor network. *Comput. Eng.* **36**(20), 113–115+118 (2010)
17. Manikandan, G., Sakthi, U.: A comprehensive survey on various key management schemes in WSN. In: 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, pp. 378–383 (2018)
18. Gautam, A.K., Kumar, R.: A comparative study of recently proposed key management schemes in wireless sensor network. In: 2018 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, Uttar Pradesh, India, pp. 512–517 (2018)
19. Bechkit, W., Challal, Y., Bouabdallah, A.: A highly scalable key pre-distribution scheme for wireless sensor networks. *IEEE Trans. Wirel. Commun.* **12**(2), 948–959 (2013)
20. Chakavarika, T.T., Chaurasia, B.K., Gupta, S.K.: Performance evaluation of a polynomial based key management scheme in wireless sensor networks. In: 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, pp. 2114–2118 (2016)
21. Kamble, S.B., Jog, V.V.: Efficient key management for dynamic wireless sensor network. In: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT), Bangalore, pp. 583–586 (2017)

22. Ahlawat, P., Dave, M.: An improved hybrid key management scheme for wireless sensor networks. In: 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, pp. 253–258 (2016)
23. Msolli, A., Ameer, H.: A new secure key management scheme for wireless sensor network. In: 2017 International Conference on Control, Automation and Diagnosis (ICCAD), Hammamet, pp. 254–257 (2017)
24. Prema, S., Pramod, T.C.: Key establishment scheme for intra and inter cluster communication in WSN. In: 2018 Second International Conference on Computing Methodologies and Communication (ICCMC), Erode, pp. 942–944 (2018)
25. Patel, J.S., Chavda, V.M.: Security vulnerability and robust security requirements using key management in sensor network. *Int. J. Grid Distrib. Comput.* **7**(3), 23–28 (2014)

Author Index

- Ahene, Emmanuel 67
- Chen, Jie 161
Chen, Yang 209
- Dai, Yusheng 228
Ding, Lin 239
- Eltayieb, Nabeil 293
- Gao, Juntao 308
Gao, Xiaoxu 21
Gervais, Mwitende 268
Gu, Dawu 239
Guan, Yuanfeng 36, 67
Guo, Ruifang 327
Guo, Yuyan 53, 145
- Han, Yanyan 347
He, Yanru 347
Hou, Jinqiu 145
Hu, Gongcheng 21, 193
Hu, Honggang 95, 128, 253
- Jia, Huiwen 107
Jiang, Mingming 53, 145
Jing, Xuan 327
- Kang, Li 3
- Li, Fagen 36, 67, 268, 293
Li, Hui 228
Li, Na 347
Li, Tuoyan 209
Li, Xuelian 308
Li, Yahong 228
Liu, Peihe 347
Liu, Ximeng 107
- Nan, Jiehui 95, 128, 253
- Qu, Quanbo 209
- Ren, Juan 177
Rong, Xing 228
- Shi, Shaoquan 84
Sun, Liang 268, 293
Sun, Mei 53
- Tian, Junfeng 327
- Wang, Chunxiao 84
Wang, Fenghe 84
Wang, Ke 268, 293
Wang, Lei 239
Wang, Leizhang 209
Wang, Qiyu 161
Wang, Yan 145
Wang, Zilong 95, 253
Wei, Shimin 53, 145
Wu, Tong 308
- Xie, Ming 53
- Yan, Xiaoxuan 347
Yin, Yifeng 107
Yu, Haiyong 308
- Zhang, Leyou 3, 21, 177, 193
Zhang, Qikun 107
Zhang, Yanhua 107
Zhang, Zhiwei 36, 67
Zheng, Mengce 95, 128, 253
Zheng, Min 228
Zhou, Yuyang 36
Zhuang, Lishuang 161