Gregorio Martínez Pérez
Sabu M. Thampi
Ryan Ko
Lei Shu (Eds.)

Communications in Computer and Information Science 420

Recent Trends in Computer Networks and Distributed Systems Security

Second International Conference, SNDS 2014
Trivandrum, India, March 13–14, 2014
Proceedings

Springer

Communications in Computer and Information Science 420

Gregorio Martínez Pérez   Sabu M. Thampi
Ryan Ko   Lei Shu (Eds.)

Recent Trends in Computer Networks and Distributed Systems Security

Second International Conference, SNDS 2014
Trivandrum, India, March 13-14, 2014
Proceedings

Springer

Volume Editors

Gregorio Martínez Pérez
University of Murcia, Spain
E-mail: gregorio@um.es

Sabu M. Thampi
Indian Institute of Information Technology
and Management - Kerala
Trivandrum, India
E-mail: smthampi@ieee.org

Ryan Ko
The University of Waikato, Hamilton, New Zealand
E-mail: ryan@waikato.ac.nz

Lei Shu
Guangdong University of Petrochemical Technology
Maoming, China
E-mail: lei.shu@lab.gdupt.edu.cn

ISSN 1865-0929 e-ISSN 1865-0937
ISBN 978-3-642-54524-5 e-ISBN 978-3-642-54525-2
DOI 10.1007/978-3-642-54525-2
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014932169

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains the papers presented at the Second International Conference on Security in Computer Networks and Distributed Systems (SNDS 2014) held in Trivandrum, India, during March 13–14, 2014. SNDS-2014 brought together students, researchers, and practitioners from both academia and industry to present their research results and development activities in the field of security in computer networks and distributed systems. The conference was organized by the Indian Institute of Information Technology and Management-Kerala (IIITM-K), Trivandrum, India, in association with ACM Trivandrum Chapter and Kerala State Council for Science, Technology and Environment (KSCSTE).

In response to the call for papers, 129 papers were submitted to the conference. These papers were evaluated on the basis of their significance, novelty, and technical quality. Each paper was reviewed by at least three members of the Program Committee. Of the papers submitted, the Program Committee selected 32 regular papers and nine short papers for presentation. In addition to the main track of presentations of accepted papers, eight papers were also presented in three co-located workshops. The authors of accepted papers made a considerable effort to take into account the comments in the version submitted to these proceedings.

There is a long list of people who volunteered their time and energy to put together the conference and who deserve acknowledgment. Thanks to all members of the Technical Program Committee, and the additional reviewers, for their hard work in evaluating and discussing papers. We wish to thank the general chairs, workshop chairs, workshop organizers, tutorial chairs, and all the members of the Steering Committee, whose work and commitment were invaluable. Our most sincere thanks go to all keynote and tutorial speakers, who shared with us their expertise and knowledge. We would like to thank all authors who submitted papers, those who presented papers, and the attendees who made this event an intellectually stimulating one. We hope they enjoyed the conference. We would also like to express our sincere gratitude to local Organizing Committee that made this event a success.

Finally, we thank Alfred Hofmann and his team at Springer-Verlag for their excellent support in bringing out these proceedings on time.

March 2014

Gregorio Martínez Pérez
Sabu M. Thampi
Ryan Ko
Lei Shu

Organized by

Indian Institute of Information Technology and Management-Kerala
(IIITM-K), Trivandrum, India

http://www.iiitmk.ac.in

Organization

Steering Committee

Ian Foster	University of Chicago, USA
Albert Y. Zomaya	The University of Sydney, Australia
Ravi Sandhu	University of Texas at San Antonio, USA
Bharat Bhargava	Purdue University, USA
Shambhu Upadhyaya	University at Buffalo, The State University of New York, USA
Gail-Joon Ahn	Arizona State University, USA
Raj Kumar Buyya	University of Melbourne, Australia
Sushil Jajodia	George Mason University Fairfax, USA
Rajasree M.S.	Director, IIITM-K, India
Peter Mueller	IBM Zurich Research Laboratory, Switzerland
Gregorio Martínez Pérez	University of Murcia, Spain
Srinivas Padmanabhuni	Infosys Labs, India and Vice President at ACM India
David Yau	Singapore University of Technology and Design and Advanced Digital Sciences Center, Singapore
Sougata Mukherjea	IBM Research, India
Vijay Varadharajan	Macquarie University, Sydney, Australia
Jerzy Konorski	Gdansk University of Technology, Poland
Chandrasekaran K.	National Institute of Technology Karnataka, India
B.M. Mehtre	Institute for Development and Research in Banking Technology (IDRBT), India

Technical Program Committee

General Chairs

Jemal H. Abawajy	Deakin University, Australia
Carlos Becker Westphall	Federal University of Santa Catarina, Brazil
Sabu M. Thampi	Indian Institute of Information Technology and Management - Kerala, India

Program Chairs

Gregorio Martínez Pérez	University of Murcia, Spain
Ryan Ko	The University of Waikato, New Zealand
Lei Shu	Guangdong University of Petrochemical Technology, P.R. China

Workshop and Special Session Chair

Ping Yang	State University of New York at Binghamton, USA

Tutorial Chairs

Pethuru Raj	Cloud Infrastructure Architect, IBM India
Sanjeev Koul	ABB Corporate Research, Bangalore, India

TPC Members

Ahmed Serhrouchni	ENST, France
Albert Levi	Sabanci University, Turkey
Alexis Olivereau	CEA LIST, France
Alf Zugenmaier	Munich University of Applied Sciences, Germany
Al-Sakib Khan Pathan	International Islamic University Malaysia (IIUM), Malaysia
Amit Sachan	Samsung Research Institute-Noida, India
Amita Sharma	I.I.S. University, India
Andrea Forte	AT&T Labs, USA
Andreas Noack	University of Applied Sciences Stralsund, Germany
Angelina Geetha	B.S. Abdur Rahman University, India
Anitha Pillai	Hindustan University, India
Anitha Varghese	ABB Research, India
Antonio Pescapé	University of Naples Federico II, Italy
Antonio Ruiz-Martínez	University of Murcia, Spain
Anyi Liu	Purdue University Fort Wayne, USA
Ashok Kumar Das	IIIT Hyderabad, India
Bernd Becker	University of Freiburg, Germany
Bhadran K.	CDAC (Centre for Development of Advanced Computing), India
Binod Vaidya	University of Ottawa, Canada
Bora Akyol	Pacific Northwest Labs, USA
Bruhadeshwar Bezawada	International Institute of Information Technology, India
Carole Bassil	Lebanese University, Lebanon
Cheng-Kang Chu	Institute for Infocomm Research, Singapore
Chunsheng Zhu	The University of British Columbia, Canada
Ciza Thomas	Indian Institute of Science, India
Debojyoti Bhattacharya	Robert Bosch Business and Engineering Solutions Ltd., India
Dija S.	Centre for Development of Advanced Computing, India

Sattar Sadkhan	University of Babylon, Iraq
Ed' Wilson Ferreira	Federal Institute of Mato Grosso, Brazil
Eric Renault	Institut Mines-Telecom – Telecom SudParis, France
Fangguo Zhang	Sun Yat-sen University, P.R. China
Feng Cheng	University of Potsdam, Germany
Geong-Sen Poh	MIMOS, Malaysia
Geyong Min	University of Bradford, UK
Ghassan Karame	NEC Laboratories Europe, Germany
Gianluca Papaleo	IEIIT, Italy
Hamed Okhravi	MIT Lincoln Laboratory, USA
Hamid Sharif	University of Nebraska-Lincoln, USA
Hong Li	Intel Corporation, USA
Jaydip Sen	Innovation lab, Tata Consultancy Services Ltd., India
Jerzy Konorski	Gdansk University of Technology, Poland
Jiangtao Li	Intel, USA
Jianhong Zhang	North China University of Technology, P.R. China
Jiannong Cao	Hong Kong Polytechnic University, Hong Kong, SAR China
Jie Li	University of Tsukuba, Japan
Joni Da Silva Fraga	UFSC, Brazil
Jordi Forne	Technical University of Catalonia, Spain
Jorge Sá Silva	University of Coimbra, Portugal
Jose Maria Alcaraz Calero	University of the West of Scotland, UK
JuCheng Yang	Tianjin University of Science and Techonology, P.R. China
Jun Wu	Waseda University, Japan
Karima Boudaoud	University of Nice Sophia Antipolis, France
Karthik Srinivasan	Philips, India
Kouichi Sakurai	Kyushu University, Japan
Kyriakos Manousakis	Applied Communication Sciences, USA
Lau Lung	UFSC, Brazil
Leonardo Martucci	Karlstad University, Sweden
Madhu Kumar S.D.	National Institute of Technology Calicut, India
Manu Malek	Stevens Institute of Technology, USA
Maode Ma	Nanyang Technological University, Singapore
Marcus Wong	Huawei Technologies, USA
Marius Marcu	Politecnica University of Timisoara, Romania
Maurizio Aiello	National Research Council, Italy
Michele Pagano	University of Pisa, Italy
Mohamed Hamdi	University of Carthage, Tunisia
Nader Mir	San Jose State University, USA

Pascal Lorenz	University of Haute Alsace, France
Patrick-Benjamin Bök	TU Dortmund, Germany
Ping Yang	Binghamton University, USA
Pradeep Atrey	University of Winnipeg, Canada
Praveen Gauravaram	Tata Consultancy Services, Hyderabad, India
Pritam Shah	Sri Venketeshwara College of Engineering Bangalore, India
Rafa Marin Lopez	University of Murcia, Spain
Rajan Ma	Tata Consultancy Services, India
Ramesh Hansdah	Indian Institute of Science, Bangalore, India
Rongxing Lu	Nanyang Technological University, Singapore
Sabrina Sicari	University of Insubria, Italy
Samir Saklikar	RSA, Security Division of EMC, India
Sara Foresti	Università degli Studi di Milano, Italy
Shambhu Upadhyaya	University at Buffalo, USA
Sherali Zeadally	University of Kentucky, USA
Sherif Rashad	Morehead State University, USA
Shiguo Lian	France Telecom, P.R. China
Shu-Ching Chen	Florida International University, USA
Skanda Muthaiah	Sandisk India Device Design Center, India
Somanath Tripathy	IIT Patna, India
Stefanos Gritzalis	University of the Aegean, Greece
Stojan Denic	Telecommunications Research Lab Toshiba, UK
Sudha Sadhasivam	PSG College of Technology, India
Sudhir Aggarwal	Florida State University, USA
Suzanne McIntosh	NYU Courant Institute of Mathematical Sciences, USA
Theodore Stergiou	Intracom Telecom, Greece
Thomas Chen	City University London, UK
Thomas Paul	TU Darmstadt, Germany
Tim Strayer	BBN Technologies, USA
Veerasamy Senthil	Thiagarajar School of Management, India
Vijayaraghavan Varadharajan	Infosys Limited, India
Vincent Roca	Inria Rhône-Alpes, France
Vinod Chandra S.S.	University of Kerala, India
Wei Yu	Towson University, USA
Weifeng Sun	Dalian University of Technology, P.R. China
Winnie Cheng	IBM Research, USA
Xianfu Lei	Utah State University, USA
Xiaoling Wu	Kyung Hee University, Korea
Xinyi Huang	Fujian Normal University, Singapore
Yassine Lakhnech	University Joseph Fourier, France

Yong Wang	Dakota State University, Madison, SD, USA
Young-Long Chen	National Taichung University of Science and Technology, Taiwan
Yu Niu	Tsinghua University, P.R. China
Yuan-Cheng Lai	Information Management, NTUST, Taiwan
Yuanfang Chen	Institut Mines-Telecom, Telecom SudParis, France
Yuexing Peng	Beijing University of Posts & Telecoms, P.R. China
Yves Roudier	EURECOM, France
Zbigniew Kotulski	Warsaw University of Technology, Poland
Zhenfu Cao	Shanghai Jiao Tong University, P.R. China
Zhenquan Qin	Dalian University of Technology, P.R. China

Additional Reviewers

Ajay Jangra	KUK University, Kurukshetra, Haryana, India
Amine Abidi	ENSI: National School of Computer Science, Tunisia
Aravind Nair	Amrita Vishwa Vidyapeetham, India
Bhagyalekshmy N. Thampi	Xlim, France
Bhawna Singla	NCCE, India
Daphne Lopez	VIT University, India
Deepak Choudhary	LPU, India
Dheerendra Mishra	Indian Institute of Technology, Kharagpur, India
Enrico Cambiaso	IEIIT, Italy
Guillaume Bouffard	Xlim, France
Komal Balasubramanian Priya Iyer	Sathyabama University, India
Kunwar Singh	NIT Trichy, India
Lalit Kumar	NIT Hamirpur (HP), India
Manjunath Mattam	International Institute of Information Technology, India
Nilanjan Dey	West Bengal University of Technology, India
Odelu Vanga	Indian Institute of Technology Kharagpur, India
Praneet Saurabh	Technocrats Institute of Technology, India
Ramalingam Anitha	PSG College of Technology, India
Rizwan Ahmed	G.H. Raisoni College of Engineering, Nagpur, India
S. Santhanalakshmi	Amrita School of Engineering, India
Saurabh Mukherjee	Banasthali University, India
Somayaji Siva Rama Krishnan	VIT University, India

T. Nishitha	JNTU, India
V. Shanthi	St. Joseph College of Engineering, India
Yatendra Sharma	Banasthali University, India

Workshop on Multidisciplinary Perspectives in Cryptology and Information Security (CIS-2014)

Workshop Organizers

Sattar B. Sadkhan	University of Babylon, Iraq
Nidaa A. Abbas	University of Babylon, Iraq

TPC Members

Alka Sawlikar	Nagpur University, India
Anil Dahiya	Manipal University Jaipur, India
Anitha Kumari	L.B.S. Institute of Technology for Women, Trivandrum, India
Archanaa Rajendran	Amrita Vishwa Vidyapeetham, India
Arsen Hayrapetyan	KIT, Germany
Atanu Rakshit	Indian Institute of Management Rohtak, India
Babu Karuppiah	Velammal College of Engineering and Technology, India
Babu Mehtre	IDRBT - Institute for Development and Research in Banking Technology, India
Balu Sridevi	VCET, India
Bhushan Trivedi	GLS Institute of Computer Technology, India
Chandra Sekaran	National Institute of Technology Karnataka, India
Chandra Vorugunti	Dhirubhai Ambani Institute of ICT, India
Deveshkumar Jinwala	S.V. National Institute of Technology, India
Dwijen Rudrapal	National Institute of Technology, Agartala, India
Gaurav Somani	Central University of Rajasthan, India
Geethakumari Gopalan Nair	BITS-Pilani, Hyderabad Campus, India
Gurudatt Kulkarni	Maharashtra State Board of Technical Education, India
Himanshu Chaurasiya	Amity School of Engineering and Technology, India
Ilaiah Kavati	IDRBT, India
Jayaprakash Kar	FCIT, King Abdulaziz University, Saudi Arabia
Jyothish Lal G.	Karpagam Institute of Technology, India
Kamatchi R.	K.J. Somaiya Institute of Management Studies and Research, India
Latesh Kumar	Siddaganga Institute Technology, India
Manjula S.H.	University Visvesvaraya College of Engineering, India

Marcus Hardt	Karlsruhe Institute of Technology, Germany
Mrudula Sarvabhatla	Sri Venkateswara University, India
Musheer Ahmad	Jamia Millia Islamia, New Delhi, India
Neelam Surti	Gujarat Technological University, India
Nirmalya Kar	National Institute of Technology Agartala, India
Prabakaran Poornachandran	Amrita Vishwa Vidyapeetham, India
Prashant Rewagad	G.H. Raisoni College of Engineering and Management, India
Priya Chandran	National Institute of Technology Calicut, India
Rajat Saxena	Indian Institute of Technology Indore, India
S. Nagarajan	Hindustan University, India
Saswathi Mukherjee	Anna University, India
Sudhish George	National Institute of Technology, Calicut, India
Swapan Debbarma	NIT Agartala, India
Syed Abbas	Ranchi University, India
Trilok Chand	PEC University of Technology, India
Venkitasamy Sureshkumar	PSG College of Technology, India
Vikas Sihag	Central University of Rajasthan, India
Vikram Raju R.	MIT, Manipal, India
Yamuna Devi	UVCE, Bangalore, India

Second International Workshop on Security in Self-organizing Networks (SelfNet 2014)

Program Chair

Al-Sakib Khan Pathan	IIUM, Malaysia

TPC Members

George Karagiannidis	Aristotle University of Thessaloniki, Greece
Nakjung Choi	Bell-Labs, Alcatel-Lucent, Korea
Abdelouahid Derhab	CERIST, Algeria
Yurong Xu	Dartmouth College, USA
Deepak Karia	Mumbai University PG Teacher, India
C-F Cheng	National Chiao Tung University, Taiwan
N.V.S.N. Sarma	National Institute of Technology Warangal, India
Ravi Kodali	National Institute of Technology, Warangal, India
Rama Rao T.	SRM University, India
Kyoung-Don Kang	State University of New York, Binghamton, USA
Axel Sikora	University of Applied Sciences Offenburg, Germany
Rui Aguiar	University of Aveiro, Portugal

Sameer Tilak	University of California at San Diego, USA
Kamran Arshad	University of Greenwich, UK
Junichi Suzuki	University of Massachusetts, Boston, USA
Giuseppe Ruggeri	University of Reggio Calabria, Italy
Liza A. Latiff	University Technology Malaysia, Malaysia
Michael Lauer	Vanille-Media, Germany

Second International Workshop on Trust and Privacy in Cyberspace (CyberTrust 2014)

Program Chair

Ashok Kumar Das	IIIT Hyderabad, India

TPC Members

Rakesh Bobba	University of Illinois at Urbana-Champaign, USA
Benoit Hudzia	SAP Research, UK
Chang Wu Yu	Chung Hua University, Taiwan
Darshan Shinde	University at Albany, USA
Deepak Garg	Thapar University, Patiala, India
Fangyang Shen	New York City College of Technology (CUNY), USA
Ghassan Karame	NEC Laboratories Europe, Germany
Guangzhi Qu	Oakland University, USA
Hamed Okhravi	MIT Lincoln Laboratory, USA
Helge Janicke	De Montfort University, UK
Houcine Hassan	Universidad Politecnica de Valencia, Spain
Jiping Xiong	Zhejiang Normal University, P.R. China
Juan-Carlos Cano	Universidad Politecnica de Valencia, Spain
Marjan Naderan	Shahid Chamran University of Ahwaz, Iran
Mohamed-Ali Kaafar	Inria, France
Phan Cong-Vinh	NTT University, Vietnam
Suzanne McIntosh	Cloudera, USA

Table of Contents

Security and Privacy in Networked Systems

Multimedia Security

Cryptosystems, Algorithms, Primitives

System and Network Security

Short Papers

Second International Workshop on Security in Self-Organising Networks (SelfNet 2014)

Workshop on Multidisciplinary Perspectives in Cryptology and Information Security (CIS 2014)

Second International Workshop on Trust and Privacy in Cyberspace (CyberTrust 2014)

Collaborative Approach for Data Integrity Verification in Cloud Computing

Rajat Saxena and Somnath Dey

Department of Computer Science and Engineering,
Indian Institute of Technology Indore, India
{rajat.saxena,somnathd}@iiti.ac.in

Abstract. High security is one of leading restriction for shining up bright eras and vision of Cloud computing. In latest trend of Cloud, all the sensitive applications and data are moved towards cloud infrastructure and data center which run on virtual computing resources in the form of virtual machine. The large scale usage of virtualization to achieve cloud infrastructure brings additional security burden for tenants of a public cloud service. In this paper, we primarily aim to achieve better data integrity verification technique and help users to utilize Data as a Service (Daas) in Cloud computing. The experimental results are included in order to show the effectiveness of the proposed method for data integrity verification.

Keywords: Proof of Retrievability (PoR), Provable Data Possession (PDP), Third Party Auditing, Algebraic Signature, Homomorphic TAG.

1 Introduction

Cloud computing is defined as services and applications that are enforced on a distributed network using virtual resources and accessed by common networking standards and Internet protocols. It is distinguished from the traditional system in this manner that resources are virtual and limitless and implementation details of the physical systems on which software runs are abstracted from the user.

In Cloud, the complexity of security is greatly increased in comparison with traditional systems. The reason for this is that data is stored and operated in multi-tenant systems which are distributed over a wider area and shared by unrelated users. In addition, maintenance of security audit logs may be difficult or impossible for a user that has limited resources. Thus, the role of cloud service providers is important that it must devote proper security measures and resources to maintain privacy preservation and data integrity. It is possible that cloud provider may delete or sell some non operational data for its greed or profit that is not used for a long time. It is also possible that an adversary may exploit this data by performing various attacks. The customer also must ensure that the provider had taken the proper security measures to protect their information.

There are a number of security threats associates with utility of DaaS in cloud computing. New security challenges introduced by storing data in the cloud are following.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 1–15, 2014.

1. **Data integrity:** when data stores on cloud storage servers, anyone from any location can access this data. Cloud is unable to differentiate between sensitive data from common data thus it enables anyone to access sensitive data. Tampering the sensitive data causes the data integrity issue. Thus, there is lack of data integrity in cloud computing.
2. **Data theft or loss:** The cloud servers are distrusted in terms of both security and reliability, which means that data may lost or modified maliciously or accidentally. Administrative errors may cause data loss (e.g. backup and restore, data migration, and changing memberships in point to point systems). Additionally, adversaries may initiate attacks for taking advantage of control loss over data by data owners.
3. **Privacy issues:** As most of the servers are external; the vendor should make sure who is accessing the data and who is maintaining the server. The cloud vendor also make sure that the customer personal information is well secured from other operators. This enables vendor to protect the personal information of customers.
4. **Infected application:** Vendor should have the complete access to the server for monitoring and maintenance. This prevents any malicious user from uploading any infected application on Cloud which will severely affect the customer.
5. **Loss of physical control:** Cloud customers have their data and program outsourced to cloud servers. As a result, owners lose direct control on the data sets and programs. Loss of physical control means that customers are unable to resist certain attacks and accidents.
6. **Data location:** In cloud environment data location are not transparent from customers. The customer doesn't know where his data is located and the Vendor does not reveal where all data is stored. The data won't even be in the same country of the customer, it might be located anywhere in the world. It might raise SLA and legal issue.
7. **Cross-VM attack via side channels:** Cross-VM attack exploits multi-tenant nature of cloud. In Multi-tenent environment, VMs belonging to different customers may co-reside on the same physical machine. Cross-VM attack may corrupt whole file structure and leak information from one VM to another VM.

We concentrate on the data integrity verification issue. It is one of the biggest concerns with cloud data storage at untrusted servers because it may be possible that cloud user or /and cloud provider may be malicious. It is also an interesting problem that how cloud users and cloud providers have trusted to each other for storing the data and how privacy of the cloud users should be maintained. One solution of this problem is to perform encryption and decryption operations but it involves with computational and operational overheads. Another solution of this problem is to perform data auditing.

Organization. The rest of the paper is organized as follows: In the section 2, we provide literature survey on data auditing till current state-of-the-art work. In section 3, we describe our system model and data integrity verification scheme.

Then, we provide security and performance analysis in section 4. Finally, we conclude in section 5.

2 Literature Survey

Data auditing is a periodic event to assess quality or utility of data for a specific purpose like to evaluate security, data integrity, privacy preservation and computational accuracy. Data auditing could be a primary source for shielding corporate data assets against potential risk and loss. Data auditing relies on a registry that could be a storage space for information and data assets. during data auditing, the creation, origin or format of data may be reviewed to assess its utility and value.

There are two type of approach for data integrity auditing: Probabilistic auditing in which the blocks are randomly selected by using the probabilistic checking method and Deterministic auditing in which auditors checks the integrity of all data blocks. Traditional systems for data auditing are PDP and PoR schemes. Both of these schemes are based on the facts that client directly communicates with the data storage to produce proof of access, retrieve and possession of the data. The difference between PDP and POR techniques is that PDP techniques only produce a proof for recoverable data possession but POR schemes checks the possession of data and it can recover data in case of data access failure or data loss. usually a PDP scheme can be transformed into POR scheme by adding erasure or error correcting codes.

The PDP techniques [1], [2], [3], [4], [5] generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. PDP techniques have two parts of action : First, the client (verifier) allows to preprocesses the data, keep a small amount of metadata and then sends whole data to an untrusted data storage server (prover) for storing. later, Client (verifier) allows to verify with the help of metadata that the data storage server still possesses the clients original data and stored data has not been tampered or deleted. In PDP techniques, the client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a low, constant amount of data that minimizes network communication. Thus, the PDP schemes for remote data checking support large data sets in widely distributed storage systems. Table 1 shows the comparative analysis of the different PDP schemes.

PoR schemes [6], [7], [8], [9], [10] have two parts of action : First, the client (verifier) allows to store a file on an untrusted data storage server or prover. later, the client run data audit proof algorithm. This proof help provers to ensure that it still possesses the clients data file and client can recover the entire file. In this schemes, an encrypted file randomly embeds a set of randomly-valued check blocks or Sentinels. The use of sentinel for data auditing minimizes the client and server storage. It also minimizes the communication complexity of the audit and the number of file-blocks accessed by server during audit. An auspiciously executed POR scheme encourages verifiers that the provers presents a protocol

Table 1. Comparison of different PDP Schemes

Properties	**PDP** [1]	**S-PDP** [2]	**E-PDP** [3]	**D-PDP**[4]	**C-DPDP**[5]
Primitives	Homomorphic Verifiable Tags (HVTs)	Symmetric key Cryptography	Asymmetric key , cryptography (RSA Modules)	Rank-based Authenticated Dictionary and Skip List, RSA Tree	Algebraic Signature
Type of guarantee	Probabilistic	Probabilistic	Probabilistic	Probabilistic	Probabilistic
Public Verifiability	Yes	No	No	No	Yes
With the help of TPA	No	No	No	No	No
Data dynamics	Append only(Static)	Yes	No	Yes	Yes
Privacy preserving	No	No	Not Supported	Not Supported	No
Support for sampling	Yes	Yes	No	Yes	Yes
Probability of detection	[1-$(1-p)^{c}$]	[1-$(1-p)^{c}$]	[1-$(1-p)^{c*s}$]	[1-$(1-p)^{c}$]	[1-$(1-p)^{c*s}$]

Table 2. Comparison of different PoR Schemes

Properties	**PoR** [6]	**C-PoR** [7]	**PoR-HA** [8]	**PoR-TI** [9]	**HAIL** [10]
Primitives	Error Correcting Code,Symmetric Key Cryptography, Sentinel Creation and Permutation	BLS Signature, Pseudorandom Functions	Error Correcting Codes,Reed -Solomon Codes Hitting Sampler	Adversarial Error Correcting Codes	Integrity Protected Error Correcting Universal Hash Function, MAC
Type of guarantee	Probabilistic	Probabilistic	Probabilistic	Probabilistic / Deterministic	Probabilistic / Deterministic
Public Verifiability	No	Yes	Yes	Yes	Yes
With the help of TPA	No	No	No	No	No
Data dynamics	No	No	Append only	Append only	Yes
Privacy preserving	No	No	No	No	Yes
Support for sampling	Yes	Yes	Yes	Yes	Yes
Probability of detection	[1-$(1-p)^{c}$]	[1-$(1-p)^{c*s}$]	[1-$(1-p)^{c}$]	[1-$(1-p)^{c}$]	[1-$(1-p)^{c*s}$]

Table 3. Comparison of different Data Auditing Techniques with TPA

Properties	**Wang et al** [11]	**Wang et al** [12]	**Hao et al** [13]	**Co-PDP**[14]
Primitives	Bilinear Map, MAC, Homomorphic Authenticator	Merkle Hash Tree, Aggregate Signature	RSA based Bilinear Homomorphic Verifiable Tags	Homomorphic Verifiable Hash Index Hierarchy
Type of guarantee	Probabilistic	Probabilistic	Deterministic	Probabilistic
Public Verifiability	Yes	Yes	Yes	Yes
With the help of TPA	Yes	Yes	Yes	Yes
Data dynamics	Yes	Yes	Yes	Yes
Privacy preserving	Yes	Yes	Yes	Yes
Support for sampling	Yes	Yes	No	Yes
Probability of detection	[1-$(1-p)^{c}$]	[1-$(1-p)^{c*s}$]	[1-$(1-p)^{c*s}$]	Z^*

1. n is the block number, c is the sampling block number and s is the numbers of sectors in blocks. p is the probability of block corruption in a cloud server and P_k is the probability of k^{th} cloud server in a multi-cloud.
2.

$$Z^* = [1 - \prod p_k \epsilon p (1-p_k)^{r_k * c * s}] \tag{1}$$

interface through which the verifiers can collectively retrieve the file. Table 2 shows the comparative analysis of different PoR schemes.

The issues with PoR and PDP schemes are: These schemes focus on only static data. These schemes apply for only encrypt files that allows a limited number of queries. There is a tradeoff between privacy preservation and dynamic data operations thus some schemes do not preserve privacy. They are complex and computation intensive and have to be done at the user end. None of this scheme consider batch auditing process. The effectiveness of these schemes primarily rests on the preprocessing steps that the user conducts before out-source the data file. This introduces significant computational and communication complexity overhead. These techniques provide tradeoff between storage overhead and cost of communication thus some of this techniques store less storage with high cost.

In cloud scenario, the users might have limited CPU, battery power and communication resource constraints. so, they are not capable to perform data audits. Instead of them, Third Party Auditors (TPA) are responsible for data audits. A trusted TPA has certain special expertise and technical capabilities, which the clients do not have.

The schemes [11], [12], [13], [14] assigns auditing work to single TPA. Trusted Third Party (TTP) involves an independent outside trusted and authenticate entity to conduct data audit. External trusted third-party audit mechanism is important and indispensable for the protection of data security and the reliability of services in cloud environment. TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data and introduce no additional on-line burden to the cloud user. The third party auditing process should bring in no new vulnerability towards user data privacy.

Table 3 shows comparative analysis of data auditing schemes that has single TPA. In these schemes, single TPA cannot handle SLA and legal issues for data possession and prone to single-point failure. For these schemes, error localization is very difficult to find. All the above schemes provide only binary results about the storage status for identifying misbehaving server(s). none of these scheme support multiple TPAs for cross checks and cross authenticate the data integrity verification, privacy preservation and computation accuracy. There is a tradeoff between data dynamics, privacy preservation and public verifiability in these schemes. TPA may simultaneously handle various audit sessions from different users for their outsourced data files by multi-user setting during efficient auditing process.

To address the above problems, we propose multiple TPA system in which each TPA may simultaneously handle various audit sessions from different users for their outsourced data files. our work utilizes the algebraic signature and homomorphic tag for auditing. Algebraic signature use symmetric key techniques to enhance efficiency. The running of algebraic signature can achieve high speed from tens to hundreds of megabytes per second. An algebraic signature allows challenger to verify data integrity by comparing only the responds returned by the storage server. for this challenger does not need whole original data for verification. Algebraic signature use only small challenges and responses. TPA group need to store only two secret keys and several random numbers. This makes task of TPA group easy and computation intensive. The efficiency of algebraic schemes permits the construction of large-scale distributed storage systems in which large amounts of storage can be verified with maximal efficiency and minimal overhead. The aggregation and algebraic properties of the algebraic signature provide extra benefit for batch auditing in our design.

By integrating the homomorphic tag with random masking, our protocol guarantees that TPA could not learn any knowledge about the data content stored in the cloud server during the efficient auditing process. Specifically, our contribution in this work can be summarized as the following three aspects:

1. We motivate the public auditing system of data storage security in Cloud Computing and provide a privacy-preserving auditing protocol with multiple TPA.
2. To the best of our knowledge, our scheme is the first to support scalable and efficient public auditing with multiple TPA in the Cloud Computing. In particular, our scheme achieves batch auditing in which each TPA may simultaneously handle various audit sessions from different users for their outsourced data files.
3. We prove the security and justify the performance of our proposed schemes through concrete experiments and comparisons with the state-of-the-art.

3 The Proposed Scheme

In this section, we present our security protocols for cloud data storage service with the aforementioned research goals in mind. first we establish notation related our scheme, then we explain details about algebraic signature. Thereafter, we discuss our system model that subsequently represent our scheme.

3.1 Notation and Preliminaries

1. AS_g $(\bullet)$: denote the Algebraic signature.
2. f$(\bullet)$ is a pseudo-random function (PRF) which maps as follow f : $\{0,1\}^k \times \{0,1\}^l \longrightarrow \{0,1\}^l$.
3. $\sigma(\bullet)$ is a pseudo-random permutation (PRP) which maps as follow σ : $\{0,1\}^k \times \{0,1........n\}$
   $\longrightarrow \{0,1.....n\}$.
4. $E_{k_t}(\bullet)$ and $D_{k_t}(\bullet)$: denote the encryption and decryption algorithms.
5. L: the length of a bit string, with typical values L = 16 bits. Each data block will be divided into equal bit strings on which the algebraic signature is computed.
6. t : the number of verification.
7. R: the number of blocks required for each challenge.
8. k: the master key, which is used to compute the locations of data blocks to compute verifiable tags.

9. k$_t$: the tag encryption key, which is used to encrypt the verifiable tags.
10. r_1, r_2: random numbers chosen from the Galois field.
11. F = F [1], F [2] . . . F [n]: F denotes a file, and F [i] denote a data block of the file F.
12. T = $T_1, T_2...T_t$: T denotes all block tags and T$_i$ denotes one tag of T.

3.2 Algebraic Signature

Algebraic signature [15] of the file blocks which has composition of strings $s_0, s_1, ..., s_{n-1}$ is a kind of hash functions simply defined as follows

$$AS_g(s_0, s_1, ..., s_{n-1}) = \sum_{s=0}^{n-1} s_i.g^i \tag{2}$$

Algebraic signature [16] itself is a single string . The data compression rate of algebraic signature is the n $=\frac{F[i]}{L}$, where F [i] is size of a file block and L is length of string. For example, if the size of a file block F [i] is 1 KB and L = 64 bits, then corresponding algebraic signature is 64 bits and n = $\frac{F[i]}{L}$ = 16, so the data compression rate of algebraic signature is 16.

Property: Taking the sum of signatures of some file blocks provides the equal result as taking the signature of sum of corresponding blocks.

$$AS_g(X) + AS_g(Y) = AS_g(X + Y) \tag{3}$$

Proof: The property of algebraic signature can be verified as follows.

$\Rightarrow \quad AS_g(X) + AS_g(Y).$
$\Rightarrow \quad AS_g(x_0, x_1, ...x_{n-1}) + AS_g(y_0, y_1, ...y_{n-1}).$
$\Rightarrow \quad \sum_{i=0}^{n-1} x_i.g^i + \sum_{i=0}^{n-1} y_i.g^i.$
$\Rightarrow \quad \sum_{i=0}^{n-1} (x_i + y_i).g^i.$
$\Rightarrow \quad AS_g(X + Y)$.

We use property of a algebraic signature for tag generation (an algebraic signature of the sum of some file blocks) which can be calculated solely using the signatures of the file blocks.

3.3 System Model

We propose a distributed multiple third party data auditing technique.In this technique, Multiple TPA have shared the huge load responsibility of single TPA by load balancing. Figure 1, illustrates network architecture for cloud data storage, which incorporates our distributed multiple third party data auditing technique. This figure divides proposed scheme into three parts:

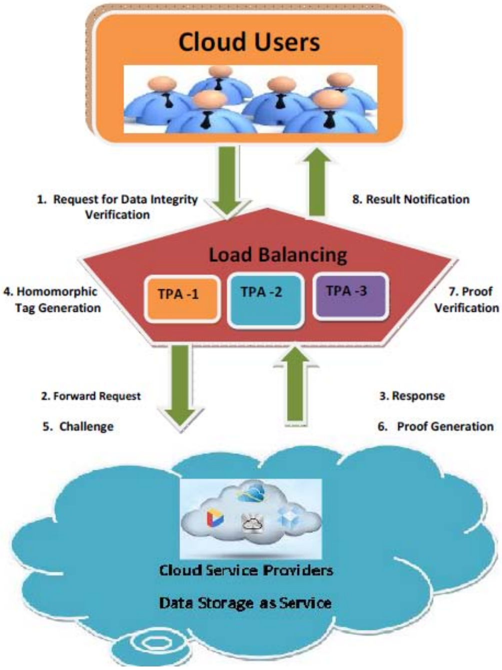


Fig. 1. System Model

1. Cloud Users: Any end user may interpreted as a cloud user. We assume that these cloud users have limited resources. Thus, Cloud user is not capable to perform computation intensive tasks such as data integrity and privacy preserving audits.

2. Multiple TPA: In this region, TPA is an authorized and authentic entity that is responsible for data integrity verification and privacy preservation. It also takes care for the SLA and legal issue related to data migration. We consider multiple TPA to achieving load balance and batch audits.

3. Cloud Service Provider: In this group, cloud service providers have established enough infrastructure and resources to provide data storage as a service for cloud customers. These resources may be distributed across the world.

3.4 Proposed Data Integrity Verification Scheme

We divide whole scheme in 8 parts and working of this parts among system model provides in Figure 1.

1. **Request for Data Integrity Verification:** During data audits, cloud users send request to the TPA group for verification of the data of file F. In TPA group one TPA can share and distribute this load with other TPA by load balancing and batch auditing.

2. **Forward Request:** TPA group use setup operation for generating some initialization parameters such as the master key k, Homomorphic Tag encryption key k$_t$ and random numbers r$_1$, r$_2$. This initial parameters are common to all TPA. TPA group forwards request to the cloud service provider on the sample blocks of physical data centers for checking data integrity.

Algorithm 1. Setup operation

```
Input: $\{0,1\}^k$ .
Output: Master key k , Homomorphic Tag Encryption Key k$_t$ , Random numbers r$_1$, r$_2$ .

1: Master key k generates by $k \overset{R}{\longleftarrow} \{0,1\}^k$.
2: Homomorphic Tag Encryption Key k$_t$ generates by $k_t \overset{R}{\longleftarrow} \{0,1\}^k$.
3: Random numbers r$_1$, r$_2$ generates by $r_1 \overset{R}{\longleftarrow} \{0,1\}^k$ and $r_2 \overset{R}{\longleftarrow} \{0,1\}^k$.
```

3. **Response:** Cloud service provider chooses some random sample blocks from the whole data base related to file F and responses to the TPA group with c blocks.
4. **Homomorphic Tag Generation:** Now, TPA group use homomorphic tag generation algorithm. Each TPA individually chooses random sample blocks $c_1, c_2, ...c_n$ from responded c blocks and compute algebraic signature sum for these blocks as the $AS_g(s_1)$, $AS_g(s_2)$, ... $AS_g(s_n)$. Through load balancing homomorphic tag generation process distribute among all TPA. TPA group sends entry (F,T) to the cloud service provider for storage.

Algorithm 2. Homomorphic TAG Generation

```
Input: Random sample blocks $c_1, c_2, ......c_n$ from responded c blocks .
Output: Entry {F , T}.

1: if the number of verification is t then
2:    TPA x has compute t tags with this procedure.
      for $0 < i \leqslant t$
            $k_i = f_k(r_1 + i)$
            $s_x = 0$
                  for 0<$j \leq c_x$
                  $l_j = \sigma_{k_i}(r_2 + j)$
                  $s_x = s_x + F[l_j]$
3:    Compute $AS_g(s_x)$
4:    $AS_g(S) = \sum_{x=1}^{n} AS_g(s_x)$ , where n is the number of TPA's.
5:    Homomorphic Verifiable Tag = $\partial_i = AS_g(S)$.
6:    $T_i = E_{k_t}(\partial_i)$.
7: end if
8: TPA group send entry {F , T}, which corresponds to file F and all block tags T,
   to the cloud service provider for store.
```

5. **Challenge:** TPA group computes k_i by challenge operation for the ith verification using the master key k, then sends the (r$_2$, k$_i$) to storage server.

Algorithm 3. Challenge operation

Input: Master Key k, Encryption function f_k and random number r$_1$.
Output: Entry {r$_2$, k$_i$ } to the storage server.

```
for $i^{th}$ verification, TPA group calculates do
    $k_i = f_k(r_1 + i)$
end for
TPA group sends {r$_2$, $k_i$ } to cloud service provider.
```

6. **Proof Generation:** In Proof Generation storage server computes the locations of the requested blocks using k_i, computes their sum F_i' and then returns to the TPA group (F_i' , T_i'), where T_i' is the homomorphic verifiable tag stored on cloud service provider corresponding to F_i'.

Algorithm 4. Proof Generation

Input: $F_i' = 0$, random number r$_2$.
Output: Cloud Service Provider return {F_i', T_i' } to TPA group.

```
$F_i' = 0$
for $0 < j \leq R$ do
    $l_j = \sigma_{k_i}(r_2 + j)$
    $F_i' = F_i' + F[l_j]$
end for
Cloud Service Provider return {$F_i'$, $T_i'$ } to TPA group.
```

7. **Proof Verification:** The TPA group decrypts T_i' using the tag decryption key k_t ,computes the algebraic signature of F_i' and then verify whether they are equal. If yes, it indicates that the integrity of file is maintained else the integrity of file is corrupted.

Algorithm 5. Proof Verification

Input: Decryption key k_t, Decryption function D_{k_t}
Output: Verification Result $AS_g(F_i') \overset{?}{=} \rho_i$

```
$\rho_i = D_{k_t}(T_i')$.
Verifies $AS_g(F_i') \overset{?}{=} \rho_i$.
```

8. **Result Notification:** TPA group notify result to the cloud user.

4 Analysis

In this section, we analyze the security strength of our scheme against server misbehavior and explain why challenge the random blocks can improve the security strength of our proposed scheme. We also provide performance analysis based on the obtained result of experiments.

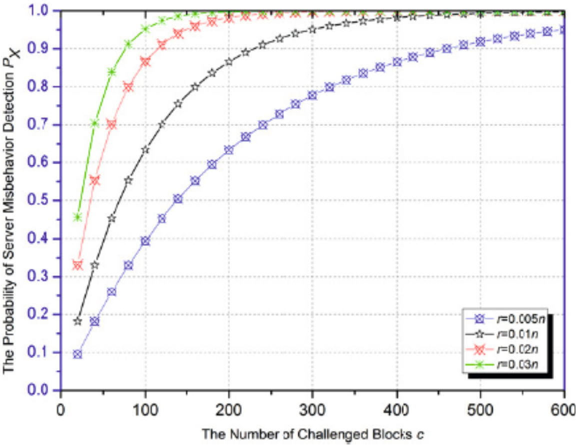


Fig. 2. The Probability of Server misbehavior detection

4.1 Security Analysis

Algebraic signature is ideally suited for use in verifying large amounts of cloud data stores in remote data centers because of their minimal network impact,reasonable computation loads and resistance to malicious modification. The use of algebraic signatures compresses file blocks into a very small entity that can change with little bit change in the block. for large bit string, Algebraic signature is cryptographically secure in comparison with hash functions such as MD5 and SHA1.

TPA group chooses c file blocks randomly from n blocks each time. This sampling incredibly reduces workload on the server, while still achieving server misbehavior detection with high provability. We assume that out of n blocks, the server deletes r blocks. X is a discrete random variable which is defined to the number of blocks chosen by TPA group that match the blocks deleted by the server. We compute P_X, the probability that at least one of the blocks picked by TPA group matches one of the blocks deleted by server, with following equation.

$$P_X = P\{X \geqslant 1\} = 1 - P\{X = 0\} = 1 - \{\frac{n-r}{n}.\frac{n-1-r}{n-1}.\frac{n-2-r}{n-2}....\frac{n-c+1-r}{n-c+1}\} \quad (4)$$

P_X indicates the probability of detection of server misbehaviour that depends on total number of file blocks n, deleted r blocks and challenged c blocks. if storage server deletes r blocks of the file, then the TPA group will detect server misbehavior after a challenge for c blocks. Figure 2 exhibit P_X for different values of r, c.

Surprisingly, the TPA group can detect server misbehaviour with a certain probability by challenge a constant amount of blocks, independently of the total number of file blocks: e.g., if r = 1% of n, then the client asks for 460 blocks and 300 blocks in order to achieve P_X of at least 99% and 95%, respectively. Thus, our scheme is probabilistic secure approach. Moreover, we can improve detection probability by performing the detection process more frequently and requesting more blocks for each challenge. Large enough bit string provides resistance from accidental collision of similar signatures. For an example, if we choose a 64 bits signature then collision probability will be 2^{-64}.

We choose 256 bits algebraic signature in our work with minimal collision probability of 2^{-256}. If the size of a file 1 GB, size of file block F [i] is 8 KB and length of bit string L = 256 bits, then corresponding algebraic signature is 256 bits and n = $\frac{F[i]}{L}$ = 256. Thus, our scheme provides data compression rate 256 for 1GB size file which is huge in cloud environment. This makes prediction task complex for a site that does not knows some secrets to generate a coherent set of signatures. The additional storage cost for 1 GB size file is = $\frac{size-of-file}{size-of-algebraic-signature}$ = 4 MB. Thus, additional storage overhead is only 4 MB for 1 GB size file.

4.2 Performance Analysis

From the performance point of view, we focuses on how frequently and efficiently user verify that storage server can faithfully store his data without retrieving it. In our scheme, the number of verification and the number of blocks required for each challenge can be set flexible according to user's requirement. If data will not be stored for a long time, user can be set small number of verification and blocks to further reduce the overload.

The experiment has run on two PCs configured with an Intel core i3-2330M 2.20 GHz and 2 GB RAM. We have configured Citrix Xen Server 6.1.0 on one PC that is used for file storage. The second PC used as a TPA group that audits the stored files on the behalf of cloud users. We observe that :

- In setup operation, TPA group generates some secret keys and random numbers.
- For Homomorphic TAG Generation, each TPA needs to perform t times PRF, c times PRP operations, c times sum, t times algebraic signature and symmetric encryption operations. If number of TPA is n, then TPA group needs to perform t*n times PRF and c*n times PRP operations, c*n times sum, t*n times algebraic signature and symmetric encryption operations.
- In challenge operation, TPA group only needs to transfer about 512 bits information for 256 bit secret keys.

- For proof generation, Cloud service provider needs to perform c times PRP and sum operations, and then it needs to transfer about 8 KB responding results (for the size block is 8KB).
- For proof verification, TPA group only needs one time decryption and comparison operations.

There only symmetric key encryption and decryption, sum, PRF and PRP operations are used. All operations are simple and efficient in computation. For our approach number of verification are infinite and server computation complexity, client computation complexity, communication complexity and client storage complexity are O(1). Thus, our approach can be utilized in cloud storage for very large data sets.

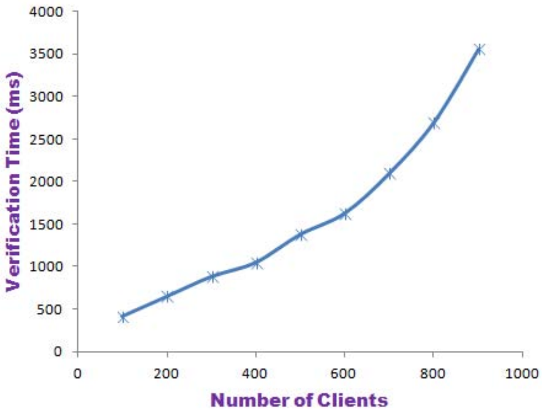


Fig. 3. The Verification delay for multiple clients

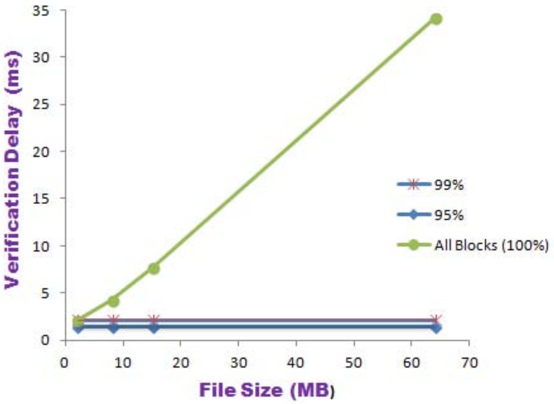


Fig. 4. The Verification overhead at multiple confidence level

We evaluate the verification delay for multiple clients in Figure 3. For 100 clients, the verification delay is about 180 ms. The delay increases quickly, when we increase the number of clients. When it reaches up to 1000 clients, the verification delay is about 3560 ms.

We measured the verification overhead for detecting 1% missing or faulty data at 100%, 99% and 95% confidence in Figure 4. Examining time for all blocks have linear scaling relationship with the file size. Sampling breaks this relationship between verification time and file size. At 99% confidence the verification overhead of our scheme is about 2.15 ms for any file. At 95% confidence the verification overhead of our scheme is about 1.4 ms for any file.

5 Conclusions and Future Work

In this paper, we propose a collective approach for data integrity verification with multiple third party auditors. This approach uses algebraic signature and homomorphic verification tag for data integrity verification. Benefits of algebraic signature and efficiency of homomorphic tag makes it ideally suited for cloud storage. Experiments shows that the performance bottleneck is bounded by disk I/O and not by our approach. Fortunately, when the server deletes a fraction of file, the client can detect server misbehavior with high probability by challenge a constant amount of blocks.

In near future, we planned to design protocols that supports dynamic data updating operations with less overhead.

References

1. Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song, D.: Provable Data Possession at Untrusted Stores. In: Proc. 14th ACM Conf. Computer and Comm. Security, pp. 598–609. ACM (2007)
2. Ateniese, G., Pietro, R.D., Mancini, L.V., Tsudik, G.: Scalable and efficient provable data possession. In: ACM SecureCom (2008)
3. Sebe, F., Domingo-Ferrer, J., Martinez-Balleste, A., Deswarte, Y., Quisquater, J.-J.: Efficient Remote Data Possession Checking in Critical Information Infrastructures. IEEE Trans. Knowledge and Data Eng. 20(8), 1034–1038 (2008)
4. Erway, C., Kupcu, A., Papamanthou, C., Tamassia, R.: Dynamic Provable Data Possession. In: Proc. 16th ACM Conf. Computer and Communication Security (CCS 2009), pp. 213–222 (2009)
5. Chen, L.: Using algebraic signatures to check data possession in cloud storage. Future Generation Computer Systems (December 2012)
6. Juels, A., Kaliski, B.S.: PORs: Proofs of retrievability for large files. In: ACM Conf. Computer and Comm. Security (2007)
7. Shacham, H., Waters, B.: Compact Proofs of Retrievability. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 90–107. Springer, Heidelberg (2008)
8. Dodis, Y., Vadhan, S.P., Wichs, D.: Proofs of retrievability via hardness amplification. In: ACM TCC-2009, pp. 109–127 (2009)
9. Bowers, K.D., Juels, A., Oprea, A.: Proofs of retrievability: Theory and implementation. In: ACM Workshop on Cloud Computing Security, pp. 43–45 (2009)

10. Bowers, K.D., Juels, A., Oprea, A.: HAIL: A high-availability and integrity layer for cloud storage. In: Proc. 16th ACM Conference on Computer and Communications Security, pp. 187–198 (2009)
11. Wang, C., Wang, Q., Ren, K., Lou, W.: Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. In: Proc. IEEE INFOCOM. IEEE (2010)
12. Wang, Q., Wang, C., Li, J., Ren, K., Lou, W.: Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing. IEEE Transactions on Parallel and Distributed Systems 22(5), 847–859 (2011)
13. Hao, Z., Zhong, S., Yu, N.: A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability. IEEE Transactions on Knowledge and Data Engineering 23(9) (September 2011)
14. Zhu, Y., Wang, H., Hu, Z., Ahn, G.-J., Hu, H., Yau, S.S.: Cooperative Provable Data Possession. In: Cryptology ePrint Archive, Report 2012/234, pp. 257–265 (2012)
15. Schwarz, T.J.E., Miller, E.L.: Store, forget, and check: using algebraic signatures to check remotely administered storage. In: Proc. of ICDCS 2006, p. 12 (2006)
16. Litwin, W., Schwarz, T.J.E.: Algebraic signatures for scalable, distributed data structures. In: ICDE 2004, Boston, MA, pp. 412–423 (2004)
17. Chen, L., Guo, G.: An efficient remote data possession checking in cloud storage. JDCTA: International Journal of Digital Content Technology and its Applications 5(4), 43–50 (2011)
18. Saxena, R., Ruj, S., Sarma, M.: Collaborative Model for Privacy Preservation and Data Integrity Verification in Cloud Computing. In: Proceedings of the Security and Privacy Symposium, IIT Kanpur, Kanpur, India, February 28-March 2 (2013)

EESOR: Energy Efficient Selective Opportunistic Routing in Wireless Sensor Networks

C.R. Yamuna Devi[1], B. Shivaraj[1], S.H. Manjula[1],
K.R. Venugopal[1], and Lalit M. Patnaik[2]

[1] Department of Computer Science and Engineering
University Visvesvaraya College of Engineering,
Bangalore, India
[2] Indian Institute of Science, Bangalore, India
yamuna.devicr@gmail.com

Abstract. Opportunistic Routing in wireless sensor networks is a multi-hop routing. In this routing neighbors of a node overhear the transmission and form multiple hops from source to the destination for transfer of information. The set of neighbor nodes participating in the routing are included in the forwarder list in the order of priority. A node with highest priority is allowed to forward the packet it hears. This paper implements Energy Efficient Selective Opportunistic Routing (EESOR), reduces the size of forwarder list by applying a condition that the forwarding node is nearer to the destination. The path followed by acknowledgment packet follows opportunistic routing, assuring reliability of transmission and energy balancing. The simulated results obtained in NS2 simulator show that proposed EESOR protocol performs better than existing Energy Efficient Opportunistic Routing (EEOR) protocol in terms of average End-to-End delay, maximum End-to-End delay and Network Lifetime.

Keywords: Delay, Energy, Forwarding, Opportunistic Routing, Wireless Sensor Network.

1 Introduction

A wireless sensor network consists of spatially distributed autonomous sensors to cooperatively monitor parameters like temperature, sound, vibration, pressure, motion or pollutants. Sensors are built by recent advances in Micro Electro Mechanical Systems (MEMS) technology. Sensory data comes from multiple sensors in distributed locations in the area where the sensor nodes are deployed. Wireless sensor networks are responsible for sensing and processing the sensed data depending on the requirement of the network. The frequency of sensing by a sensor node may depend on the occurrence of an event or it may be periodical depending on the application for which a node is used.

Industrial application areas where wireless sensor nodes are used include Industrial process monitoring and control machines. Health monitoring, environment and habitat monitoring, health care applications, home automation, and traffic control, are civilian applications of wireless sensor networks. The applications of wireless sensor

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 16–31, 2014.

networks demand smaller size for the nodes, and in turn, the components of the nodes. The lifetime of a sensor node depends mainly on the battery contained in the nodes. Because of the wireless nature of nodes, the applications demand long life for sensor nodes. This requires energy of the sensor nodes to be used very efficiently. Applications of wireless sensor networks demand minimum delay in data transmission, good throughput, and longer network lifetime.

Figure 1 shows the block diagram of a wireless sensor node. A typical wireless sensor node has a sensor unit to perform the basic sensing operations, a memory unit to store the sensed data, a battery for power requirements, an embedded processor for data processing and a transceiver for transmitting and receiving the data. The transceiver unit is provided with a limited range antenna. The memory provided in the sensor node is normally a limited storage memory to facilitate small size, as the node is transmitting the data sensed and does not store it. The battery is provided with initial energy and has limited life. With the five necessary units a sensor node can be equipped with optional units such as mobilizer, location finding unit and power generator. Normally, about 50% of the cost of the sensor node is meant for the cost of the actual sensor unit.

The design of a routing protocol for wireless sensor network is influenced by factors like *scalability, fault tolerance, network topology, transmission media, operating environment and power consumption*. It is difficult to integrate all these factors into a single network and they are only used as guidelines in the design of a routing protocol. The influencing factors are used for the comparison of different routing schemes for wireless sensor networks. Many applications in wireless sensor networks require information to be transferred between source-destination pairs, that may be one or more hops away.

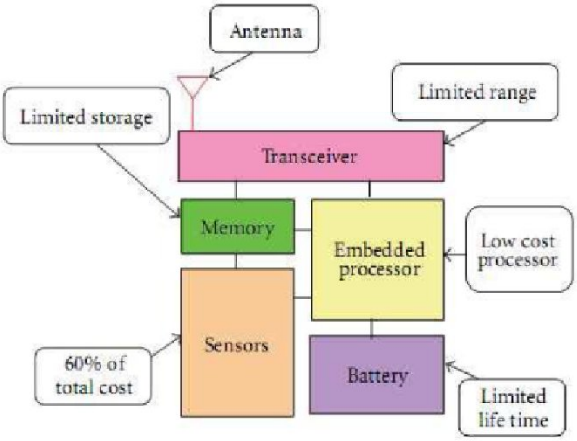


Fig. 1. Wireless Sensor Node Block Diagram

It is an interesting problem to connect the source-destination pairs of a wireless sensor network through the shortest distance, with minimum hops, in a short time with more reliability. When the source and destination are more than one hop away, one of the nodes has to be selected from the set of neighbours of the source to forward the packet towards the destination. The nodes in the forwarder list are prioritized based on different metrics like hop count and packet delivery ratio. The choosing of forwarding node continues till the destination node is reached. Different routing protocols are used in disseminating information from source to destination in a wireless sensor network.

Opportunistic routing is one of the flat based reactive routing protocols. Advantages of opportunistic routing protocols are increased reliability and increased transmission range of a node in a wireless sensor network. Network reliability is increased by transmitting a packet through any possible link in the network rather than one specified link. Transmission range is increased by including good quality short-ranged links and poor quality long-ranged links. There are many variants of opportunistic routing, *viz*, Energy Efficient Opportunistic Routing (EEOR) [1], Exclusive Opportunistic Routing (ExOR) [2], Assistant Opportunistic Routing (AsOR) [3]. Each of the opportunistic routing protocol has its own advantages and disadvantages.

Energy Efficient Opportunistic Routing is a multi-hop routing protocol for wireless sensor networks. It makes use of the forwarders list of the node to choose the forwarding node to transfer the data towards the target. Priorities are assigned for the neighbours of a node to choose the forwarding node. Energy consumption, packet loss ratio, and delivery delay parameters in a wireless sensor network are measured. Efficient protocols are required to reduce delay in transmission and to prolong the network lifetime. EEOR protocol gives better results compared to ExOR protocol in terms of packet loss ratio, average delivery delay and energy consumption. The proposed EESOR protocol achieves better throughput, maximum end-to-end delay and network lifetime, by reducing the size of the forwarder list and opportunistically routing the acknowledgment packet from target to source in the network.

The rest of the paper is organized as follows. In Section 2, we discuss the work related to opportunistic routing. The Background required for the design of a new protocol for wireless sensor network is discussed in Section 3. Section 4, we present the assumptions and implementation details of Energy Efficient Selective Opportunistic Routing. We analyze the newly implemented protocol for different performance parameters in Section 5. Finally, we conclude the paper and discuss future scope in Section 6.

2 Related Work

A vast amount of literature is devoted to opportunistic routing in wireless sensor networks. Some of important works are discussed below.

Mao *et al.,* [1] focused on selecting and prioritizing the forwarder list of a node to minimize energy consumption by all the nodes by designing Energy Efficient Opportunistic Routing (EEOR). The network is analyzed for energy consumption, average delivery delay and packet loss ratio. Biswas and Moris, [2] proposed ExOR, to describe an integrated routing and MAC protocol that increases the throughput of multi-hop wireless networks. The protocol chooses each hop destination of a packet's route after the completion of the hop. It gives the choice to decide which of the neighbouring nodes receives the packet. ExOR protocol gives a better throughput compared to traditional routing with the same network capacity. Wei *et al.,* [3] discussed Assistant Opportunistic Routing (AsOR) protocol, that is a unicast routing protocol for multi-hop wireless sensor networks. The authors provided a method for the optimal value for the number of nodes in one transmission segment for multi-hop networks.

Bhorkar *et al.,* [4] proposed d-AdaptOR, a distributed, adaptive, and opportunistic routing scheme for multihop wireless ad hoc networks. The advantage of this scheme is optimal performance, with zero knowledge of network topology and channel statistics. The disadvantage of this protocol is ignorance of short-term performance and congestion control in the network. Hsu *et al.,* [5] reviewed the basic concepts of opportunistic routing and describes the components of opportunistic routing. Current trends, issues and challenges in opportunistic routing are explained with examples. The authors discussed the key challenges in opportunistic routing to be addressed by the researchers.

Wang *et al.,* [6] provided a solution for quality transmission, with the combination of link quality variation and broadcasting nature of wireless channels. The solution called CORMAN, gives significant performance improvement over Ad hoc On-demand Distance Vector routing, with varying mobile settings. Chachulski *et al.,* [7] presented MORE, a MAC-independent opportunistic protocol, that randomly mixes packets before forwarding them. This protocol runs directly on top of 802.11, without the need for any special scheduler. MORE is found to be better than ExOR, with respect to throughput and gain, when there is spatial reuse, both for unicast and multicast. Fang *et al.,* [8] discussed the problems of choosing the opportunistic route to optimize the total utility or profit of multiple simultaneous users in a wireless mesh network. CONSORT, node-CONStrained Opportunistic RouTing, is proposed by combining the primal-dual and subgradient methods. The iterative method reduces the gap between the solutions provided in each iteration and provides the optimal solution, with respect to higher user utilities and profits.

Mazumdar and Sairam, [9] discussed the opportunities and challenges in opportunistic routing, in improving the performance of wireless multi-hop Ad-hoc and wireless sensor networks. Further, the paper outlines several vital design issues that needs to be considered in improving efficiency and deployability of networks. Li *et al.,* [10] proposed the Localized Opportunistic Routing (LOR) protocol that utilizes the distributed minimum transmission selection algorithm to partition the topology into several nested close-node-sets using local information. LOR improves performances over extremely opportunistic routing and MAC-independent opportunistic routing protocol considering the parameters control overhead, end-to-end delay and throughput. Rozner *et al.,* [11] developed a model-driven optimization framework to optimize opportunistic routes and rate limits for both unicast and multicast traffic. The authors conducted simulations and test bed experiments to show that the proposed algorithm outperforms shortest path and opportunistic routing protocols. Traffic and topology variations are to be addressed in the future work.

Nasipuri *et al.,* [12] developed algorithms to find the path that consumes minimal energy for node and link-disjoint wireless networks. The performance evaluation shows that the link-disjoint paths consume less energy than node-disjoint paths. The issues related to distributed implementation and optimal centralized algorithms are discussed. Basalamah *et al.,* [13] analyzed opportunistic routing gain under link correlation with the loss of data and acknowledgment packets. A new link-correlation-aware opportunistic routing scheme is introduced, which exploits the diverse uncorrelated forward links. The simulation work captures the full advantage of opportunistic routing. Levorato *et al.,* [14] have used Multiple-Input Multiple-Output technology, to increase communication parallelism in the network, by multiple concurrent information flows. A cooperative cross-layer scheme is proposed integrating distributed incremental redundancy hybrid automatic retransmission request error control with routing. The scheme to improve the efficiency of transmissions and reduces the interference in the network. Results in the paper show that the network performance is significantly increased.

Zeng *et al.,* [15] gives a comprehensive study on the impacts of multiple rates, interference, and prioritization on the maximum end-to-end throughput and capacity of opportunistic routing. It is shown that opportunistic routing has a higher potential to improve end-to-end throughput. Wang *et al.,* [16] suggested two opportunistic routing algorithms for P2P networks, that exploit the spatial locality, spatial regularity and activity heterogeneity of mobile nodes in a network. Both theoretical analysis and simulation based study reveal that the proposed algorithms outperform the other algorithms in terms of delivery latency and delivery ratio. Shin *et al.,* [17] proposed a parallel opportunistic routing for wireless ad-hoc networks to observe the changes in power, delay and throughput as the number of source-destination pairs increases in the network. A net improvement in overall power-delay trade off is seen as compared to conventional routing since the interference tolerance of receivers is increased in the network.

Passarella *et al.,* [18] investigated the use of a series of opportunistic contacts, in an opportunistic routing environment. The authors implement a scheme for supporting service provisioning in opportunistic networks, and an analytical model for determining the optimal number of parallel executions required to minimize the service time in the network. Myung and Lee [19] proposed a method for avoiding duplicate forwarding of packets in opportunistic routing. The packet includes a small information piggybacked, that reduces the number of repeated packet transmissions, that in turn increases the throughput.

3 Background

Routing Protocols aim at improving the performance of wireless sensor networks, in terms of the lifetime, the delay and the network throughput. Routing protocols are classified as single hop and multi hop networks, depending on the number of hops to connect the source and target in the network. Based on the network structure routing protocols are classified as flat based, cluster based and location based routing protocols. The establishment of routing path in a wireless sensor network gives reactive and proactive routing protocols. Opportunistic routing is a flat based, reactive, multi-hop routing protocol for wireless sensor networks which applies to both small scale and large scale wireless sensor networks. Opportunistic routing exploits the broadcast nature of wireless sensor networks.

The metrics used for forwarder set selection are hop-count, packet delivery ratio and end-to-end delay. Any one or a combination of these metrics can be used in forwarder set selection and for prioritizing the forwarder nodes. Opportunistic routing has potential benefits brought to wireless sensor networks. Challenges faced by opportunistic routing are taken based on network coding coordination, multi-flow rate control, power control with proper bit-rate selection, multi-channel scenario, deployment of nodes and combination of opportunistic routing with selection diversity. Opportunistic routing is analyzed for fixed power model and adjustable power model of wireless sensor network.

ExOR [2] multi-hop routing for wireless sensor networks integrates routing and MAC protocol to increase the throughput of multi-hop wireless sensor networks. ExOR uses long radio links with high loss rates for transfer of data in a network, that usually are avoided in traditional routing. With the same network capacity as traditional routing, ExOR results in high throughput in a multi-hop wireless sensor network. AsOR [3] forwards the data from the source to destination through a sequence of intermediate nodes. The assistant nodes are used to provide protection for unsuccessful opportunistic transmissions. Priority is given to conservation of energy in wireless sensor networks in the implementation of AsOR.

EEOR [1] allows one of the neighbours to participate in the forwarding of the data packets, from source to destination in multi-hop transmissions in a wireless sensor network. The forwarding node is chosen depending on the cost assigned to each of the nodes. To handle the network traffic efficiently, congestion is controlled in the network dynamically adjusting the flow from each source node in the network. Penalty is imposed on the nodes who choose more than one forwarder nodes in multi-hop transmission. The protocol computes the expected cost for each node and selecting the forwarder list. From the forwarder list the optimal forwarder list is found by opportunistic routing. The proposed EESOR performs better than the existing EEOR in terms of average End-to-End delay, maximum End-to-End delay and network lifetime.

4 Problem Definition

Wireless sensor network can be single or multi-hop network depending on the transmission range of the sensor nodes. More number of hops increases the delay in transmission and energy consumed by the nodes. The objective of this work is to reduce the energy consumed by the sensor nodes in receiving, transmitting of information and to decrease the delay in transmission of data from source to destination in a wireless sensor network.

The following are the assumptions in the wireless sensor networks considered:

1) The sensor nodes deployed randomly are static.
2) The node density in the network is uniform.
3) All the sensor nodes are deposited with same initial energy.
4) The sensory data generation frequency is uniform.
5) The nodes have same transmission range.

Sending a packet from source to target in a network can be considered to include three parts, *viz* 1) the source sending the packet to one neighbour node and that node is the target node, 2) if the target is more than one hop away from the source, then there is at least one node in the neighbours list to relay the packet to target, and

3) agreement on choosing the actual relay node, among the neighbours of the transmitting node. The time and effort incurred achieving the part 1, is constant. The same for part 2 depends on the distance between the source and the destination. It is very hard to find the cost on coming to an agreement as to choose the relaying node. It is assumed that the overall cost of communication is represented by the distance between the nodes to be communicated in the wireless sensor network. The distance d between two nodes $A(x1,y1)$ and $B(x1,y2)$ is calculated by the equation,

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \tag{1}$$

Table 1. Network Parameter Notations

Variable	Description
N	number of the nodes in the network
P	number of the packets transmitted between a pair of source and destination
X, Y	Maximum value for node's position
T_r	Transmission Range
n_x	x co-ordinate of node n
n_y	y co-ordinate of node n
d	Time taken by a packet in reaching destination from the source node
E_I	Initial energy of the node
E_c	Critical energy level of the node
E_r	Residual energy of the node
T	Simulation time

Let s represent the source node, t represents the target node and $i_1, i_2, .., i_n$ represent the intermediate nodes in the network. $F(n)$ represents set of forwarder nodes for node n. The set of sorted forwarder nodes of n is represented by $FS(n)$. It is obvious that $FS(n) = i_1, i_2, .., i_n$ probably in a different order. Let α represent the probability that a packet sent by node s is not received by any of the nodes in $FS(n)$. Then β represents the probability that a packet sent by node s is received by at least one node in $FS(n)$, which implies the equation

$$\beta = 1 - \alpha \tag{2}$$

The Table 1 shows the list of variables used in the network and their notations. The network scenario can be represented by the mathematical model shown below.

Optimization function is to

$$\text{Minimize } d \tag{3}$$

Subject to the following constraints

$$E_r <= E_I \text{ for all nodes;} \tag{4}$$

$$E_r <= E_c \text{ ; for all live nodes;} \tag{5}$$

$$0 <= n_x <= X \text{ ; } 0 <= n_y <= Y; \tag{6}$$

In the network considered, the source node forms the set of neighbouring nodes to forward the packet, when the destination is more than one hop away from the source. The set of neighbours is sorted according to its distance from the destination, and normally the first of these nodes in the forwarder list relays the packet towards the destination. The procedure continues till the destination node receives the packet. Algorithm Energy Efficient Selective Opportunistic Routing is given in Table 2 This algorithm finds the minimal path between the source and destination pairs specified in the network.

Table 2. Algorithm EESOR: Energy Efficient Selective Opportunistic Routing

Input: Randomly deployed sensor nodes with source and destination pair to be connected.

Output: Path between source-destination pairs with minimal hops.

Step 1: Construct the routing table for all nodes.

Step 2: Form the neighbor list of each source node.

Step 3: Sort it according to ascending order of distance between itself and destination.

Step 4: Relay the data to first node in the sorted list.

Step 5: Update the routing table of the forwarding node.

Step 6: If destination is reached stop else repeat steps 2-4.

Step 7: Transmit acknowledgment towards the source using steps 1-4.

Step 8: Repeat steps 1-5 for all the source nodes in the network.

The fields used in the packet to be communicated are shown below.

1) source
2) packet length
3) packet sequence number
4) *x* coordinate
5) *y* coordinate
6) *z* coordinate
7) distance
8) data

The first field source represents the node that originated the packet. Packet length represents the number of bytes contained in the packet. Packet sequence number is the index of the packet in the overall simulation of the network. *x* and *y* coordinates represents the position of the node, and *z* represents the speed of movement of a mobile node in number of steps per second. As the nodes in our network are static, *z* coordinate is always 0. The distance field of the packet represents the geographical distance between the node and the source. The last field is the data to be communicated between the source and the destination nodes. The acknowledgment packet has the same fields, except the data field.

The routing table of a node consists of the following fields: *Destination, Next Hop, Packet Sequence Number and Distance from the node to destination.* Each node has a routing table of all its neighbours, consisting of all the required fields. Distance between node and target node is used in updating the routing table entry of the node during multi-hop transmission. Construction of routing table of a node is considered in the following phases. In Phase I, the routing table for all the nodes in the wireless sensor network is constructed. Phase II updates the routing table of the forwarding nodes depending on their distance from the target node in multi hop wireless sensor networks.

4.1 Phase I - Populating the Routing Table

Initially, the routing table of every node is constructed based on the neighbouring node information. This phase starts with the construction of a *HELLO* packet from the node to all its neighbours. Once it is done, a timer is used to *broadcast* the *HELLO* packets to all of its neighbours. The *HELLO* packet is not sent to any particular node. The node that receives packet, checks the packet source field to find out the address of the node that originated the *HELLO* packet. If the receiver node routing table already has an entry of the source node of the *HELLO* packet, it drops the packet. Otherwise, it creates a new entry for the node that has sent the *HELLO* packet with all the necessary fields.

This process of sending *HELLO* packets and updating the routing table entry, if needed, is continued for all the nodes in the networks have the complete routing table entries. At the end of Phase I, each node is having a routing table for itself, containing all the information of its immediate neighbouring nodes.

4.2 Phase II - Updating the Routing Table on the Fly

The routing table constructed in Phase I, is sufficient for communication in a network, if the nodes are one hop distant away only, that is not true always. In Phase II, the entries in the routing table are updated, depending on the position of the source, intermediate nodes and the destination. The distance between the forwarding node and the target node is updated in the routing table entry, so that the next hop node is the one with smallest distance between itself and the target node. According to the concept of Energy-Efficient Selective Opportunistic Routing (EESOR), the next hop to a particular destination is decided on the fly and new protocol implemented is completely opportunistic. This process is repeated till the destination node is reached. At the end of Phase II, we have a shortest path from the source to destination for multi-hop paths in a wireless sensor network.

The Transport Layer Protocol used in this communication is Transmission Control Protocol (TCP). TCP is a reliable protocol, where every packet is guaranteed to be delivered to the destination, by making use of the acknowledgment packet, sent from target to source node. Normally, the acknowledgment packet flows in the reverse path of the data path, using same intermediate nodes. The newly implemented protocol

Energy Efficient Selective Opportunistic Routing sends the acknowledgment packet opportunistically.

Figure 2 shows an example multi-hop transmission in a wireless sensor network with 10 nodes randomly deployed. Consider the transmission between the source-destination node pairs 8 and 3. The existing EEOR protocol takes the path through the intermediate nodes 7, 2 and 9, before reaching the destination node. The acknowledgment packet flows in the reverse path. EESOR protocol takes the path through 2 and 9, avoiding node 7, as shown by dark lines and the acknowledgment flows through the nodes 9 and 7, indicated by light lines avoiding the same route as the data packet in the Fig. 2. Real time communication involves transmission between different pairs of nodes at different instances of network functioning and the intermediate nodes are chosen according to steps of the algorithm. This real time communication and the use of different paths for the data and acknowledgment packets balances the energy consumed by the nodes in the network and prolongs the network lifetime.

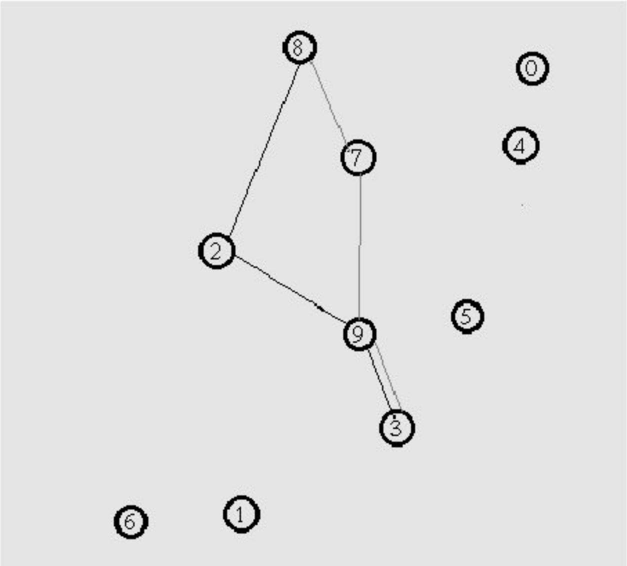


Fig. 2. Energy Efficient Selective Opportunistic Routing for data and acknowledgment packet

5 Performance Evaluation

5.1 Simulation Setup

The simulator used in the analyzing the wireless sensor network in this paper is NS2. This section provides simulation setup to demonstrate performance of Energy Efficient Selective Opportunistic Routing in the wireless sensor networks. 50 wireless sensor nodes are deployed randomly in a square area of 500m by 500 m, with uniform distribution. The packet generation rate is one packet per second. Packets of 1000 bytes each, are transferred between source and destination pairs for a simulation time of 150 seconds. The acknowledgment packet size is 40 bytes. All the sensor nodes in the network are deposited with an initial energy of 50 Joules. The energy spent by a sensor node in transmission of packet is maximum of 0.38 Joules, in receiving is 0.36 Joules. The node consumes a minimum energy of 0.003 Joules, when it is in idle state.

The behaviour of the network is observed for average End-to-End delay, maximum End-to-End delay and network lifetime. In these 50 nodes, 9 different source- destination pairs are randomly chosen for one-hop, two-hop, and more than two-hop communications. Table 3 shows the details of the nodes connected in the wireless sensor network. The node-pairs 4-41, 43-34, and 39-40 represent one hop communication pairs. The node-pairs 30-32, 10-20, and 1-5 represent two hop communication pairs. The last pairs 0-24, 11-22, and 33-49, represent more than two hops communication source-destination pairs.

Table 3. Details Of Connected Pairs In The Network

Number of Hops	Source Node	Destination Node
1	4	41
1	43	34
1	39	40
2	30	32
2	10	20
2	1	5
More than 2	0	24
More than 2	11	22
More than 2	33	49

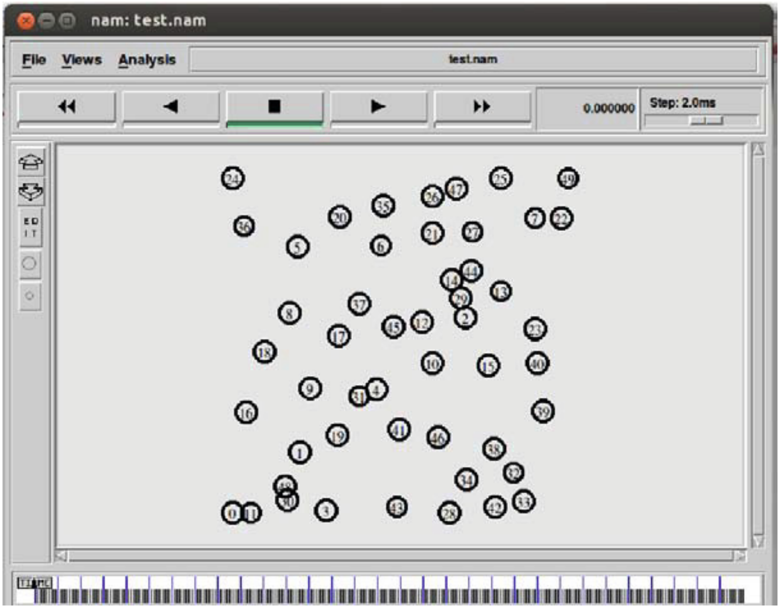


Fig. 3. Network Scenario with 50 Nodes

For one hop communication, it is not necessary to find the forwarder list and sort it. For two hop communications, the neighbour list of the source has to be formed, prioritized and sorted according to their distance from the target. The node in the forwarder list that is nearer to target is chosen as the forwarding node. For the pairs, 30-32, 10-20 and 1-5, the nodes 28, 35 and 8 are used as relaying nodes, respectively. For more than two hop node-pairs, 0-24, the nodes 18 and 36 are used as relaying nodes. For the next pair 11-22, data from 11 goes to node 31, then from node 31 to node 14, and from 14 it reaches the target 22. Nodes 40 and 22 are used as relaying nodes in connecting the last pair 33-49.

5.2 Performance Analysis

This section analyzes the performance of the wireless sensor network for Energy Efficient Selective Opportunistic Routing for the parameters Maximum End-to-End delay, average End-to-End delay and network lifetime. Fig. 3 shows the network scenario of the wireless sensor network with 50 nodes randomly deployed in the area of 500 m X 500 m. 250 m is the transmission range of each of the sensor nodes in the network.

5.2.1 Average End-to-End Delay

End-to-End Delay is defined as the time elapsed between the source node sending the packet and the destination node receiving the packet. The average of the End-to-End delay of all the packets transmitted between each of the pairs of source-destinations gives the average End-to-End delay. The average End-to-End delay is plotted against different pairs of source and destinations as shown in Fig. 4.

It is observed that for one hop networks, Energy Efficient Selective Opportunistic Routing does not show any improvement, because the time for choosing the set of forwarder list is not needed. As the number of hops increases, the reduction in delay is more. For two-hop and three-hop communications the delay is reduced up to a maximum of 90 ms which is the same as 9% for 10-20 source-destination pair and 295 ms that is approximately equal to 30 %for 11-22 source destination pair, respectively.

5.2.2 Maximum End-to-End Delay

Figure 5 shows the plot of maximum of End-to-End delay values, for the same 9 pairs of nodes considered for analyzing average End-to-End delay. Once again, single hop communication takes same amount of time in Energy Efficient Selective Opportunistic Routing. Two-hop communication between the nodes 30 and 32 shows the maximum improvement of around 300 ms, or 3 % of total delay for each source destination pair. And more than two-hop communication yields a maximum reduction of delay by approximately 1000 ms, or 50 %, for the source destination pair 11-22. The reason for this reduction is decrease in the size of forwarder list in case of Energy Efficient Selective Opportunistic Routing, by considering only the neighbour nodes that are nearer to destination.

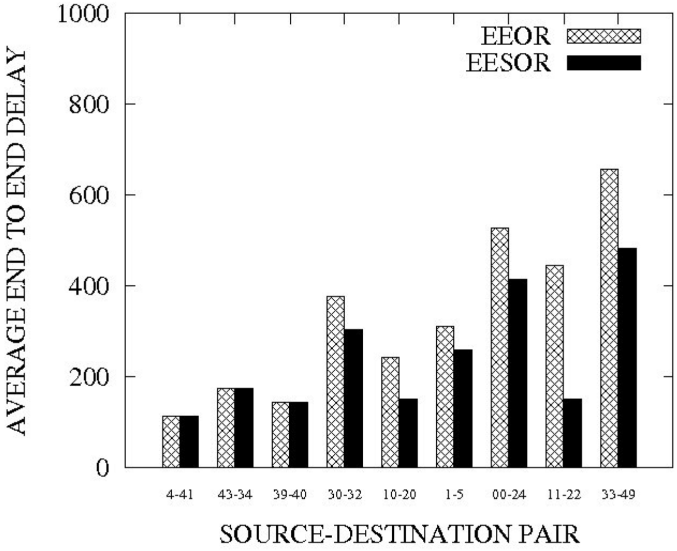


Fig. 4. Average End-to-End Delay

The small size of forwarder list reduces the time taken for prioritizing and sorting the nodes. The time for finding the shortest distance between each node in the forwarder list and the destination is reduced. The analysis of maximum End-to-End delay shows that, as the number of hops increases, the transmission delay increases. The End-to-End delay is lesser in Energy Efficient Selective Opportunistic Routing as compared to Energy Efficient Opportunistic Routing.

5.2.3 Network Lifetime

The *lifetime* of a sensor node is considered as the time from its deployment to the time till which the node is having more than 10% of its initial energy. The node is said to be *alive* in this period. Beyond this period the node is said to be *dead. Network Lifetime* is the time between inception of the network to the time upto which 10% of the sensor nodes are alive. Fig. 6 shows the network lifetime for both Energy Efficient Opportunistic Routing and Energy Efficient Selective Opportunistic Routing protocols plotted against different network sizes. Network size is considered as 25 nodes, 50 nodes, 75 nodes and 100 nodes for comparing the performance of Energy Efficient Opportunistic Routing and Energy Efficient Selective Opportunistic Routing.

The network performance is analyzed for the network sizes 25, 50,75 and 100 nodes for network lifetime. The graph shows that the network lifetime increases for all the networks considered, irrespective of the network size. The reason is Energy Efficient Selective Opportunistic Routing uses lesser number of hops to reach destination from the source and the acknowledgment packet traverses a path that may not be the same as the data path as compared to Energy Efficient Opportunistic Routing.

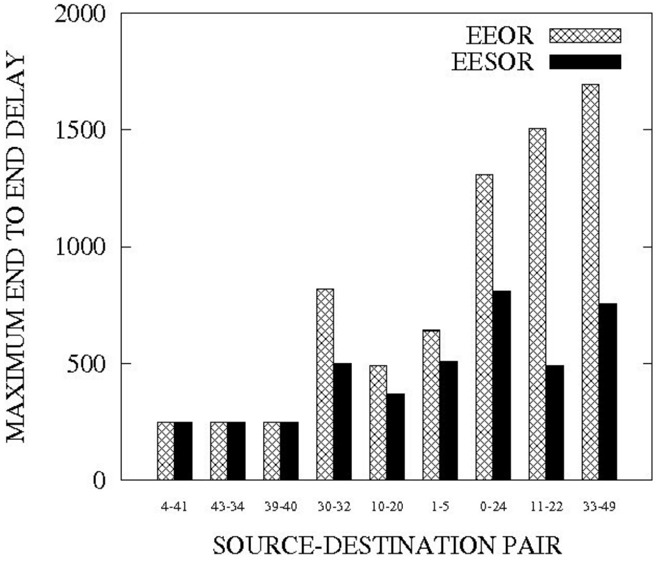


Fig. 5. Maximum End-to-End Delay

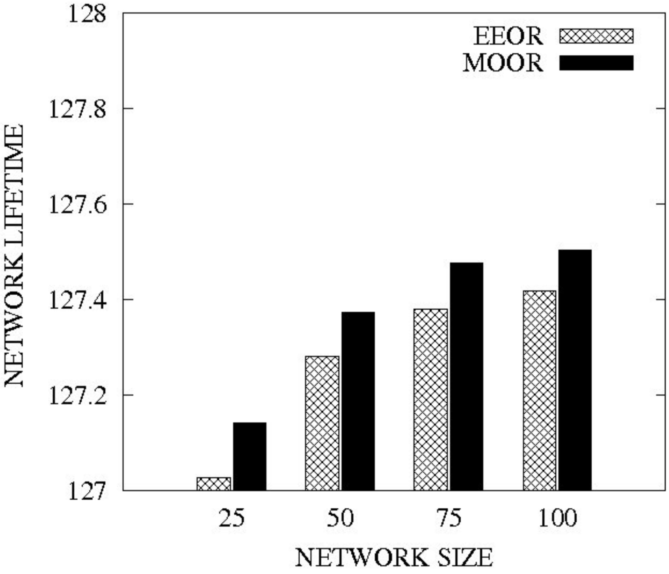


Fig. 6. Network Lifetime

6 Conclusions

In this paper, a novel approach, Energy Efficient Selective Opportunistic Routing is presented, to reduce the average end-to-end delay, maximum end-to-end delay and to increase the lifetime of a multi-hop wireless sensor network. The size of set of forwarder nodes of the source node is reduced by imposing a condition that, the neighbour nodes of the source node nearer to the destination are selected to be included in the set of forwarder nodes. These nodes in the forwarder list are sorted according to the descending order of their distance from the destination node. This technique decreases the average End-to-End delay and maximum End-to-End delay up to maximum of 50% for some of the source and destination pairs as compared to existing Energy Efficient Opportunistic Routing. Opportunistic routing is applied for the flow of acknowledgment packet from target to source, to balance the energy consumption among the nodes in the network. The lifetime of the network is increased compared to Energy Efficient Opportunistic Routing. The use of Transmission Control Protocol results in the packet delivery ratio to be almost cent percent, as the protocol is reliable. A future enhancement is to analyze Energy Efficient Selective Opportunistic Routing for performance parameters like throughput and turn around time in wireless sensor network.

References

1. Mao, X.F., Tang, S., Xu, X., Li, X.Y., Ma, H.: Energy Efficient Opportunistic Routing in Wireless Networks. Proceedings of IEEE Transactions on Parallel and Distributed Systems 22(11), 1934–1942 (2011)
2. Biswas, S., Morris, R.: ExOR: Opportunistic Multi-hop Routing for Wireless Networks. In: Proceedings of ACM SICGOMM, pp. 133–144 (2005)
3. Wei, C., Zhi, C., Fan, P., Letaief, K.B.: AsOR: An Energy Efficient Multi-Hop Opportunistic Routing Protocol for Wireless Sensor Networks over Rayleigh Fading Channels. IEEE/ACM Transactions on Wireless Communications 8(5), 2452–2463 (2009)
4. Bhorkar, A.A., Naghshvar, M., Javidi, T., Rao, B.D.: Adaptive Opportunistic Routing for Wireless Ad Hoc Networks. IEEE/ACM Transactions on Networking 20, 243–256 (2012)
5. Hsu, C.-J., Liu, H.-I., Seah, W.K.G.: Opportunistic Routing A Review and the Challenges Ahead. ELSEVIER Journal Computer Networks S5, 3592–3603 (2011)
6. Wang, Z., Chen, Y., Li, C.: CORMAN: A Novel Cooperative Opportunistic Routing Scheme in Mobile Ad Hoc Networks. IEEE Journal on Selected Areas in Communication 30, 289–286 (2012)
7. Chachulski, S., Jennings, M., Katti, S., Katabi, D.: Trading Structure for Randomness in Wireless Opportunistic Routing. In: Proceedings of ACM SIGCOMM (2007)
8. Fang, X., Yang, D., Xue, G.: CONSORT: Node constrained Opportunistic Routing in Wireless Mesh Networks. In: Proceedings of IEEE INFOCOM, pp. 1893–1898 (2011)
9. Mazumdar, A.P., Sairam, A.S.: Opportunistic Routing: Opportunities and Challenges. International Journal Information and Electronics Engineering 2, 247–252 (2012)
10. Li, Y., Mohaisen, A., Zhang, Z.-L.: Trading Optimality for Scalability in Large-Scale Opprotunistic Routing. IEEE Transactions on Vehicular Technology 62(5), 2253–2263 (2013)

11. Rozner, E., Han, M.K., Qiu, L., Zhang, Y.: Model-Driven Optimization of Opportunistic Routing. IEEE Transactions on Networking 21(2), 594–609 (2013)
12. Nasipuri, A., Castaneda, R., Das, S.R.: Performance of Multipath Routing for On-Demand Protocols in Ad Hoc Networks. ACM/Kluwer Mobile Networks and Applications 6(4), 339–349 (2001)
13. Basalamah, A., Kim, S.M., Guo, S., He, T., Tobe, Y.: Link Correlation Aware Opportunistic Routing. In: Proceedings of 31st Annual International Conference on Computer Communications, pp. 3318–3322 (2012)
14. Levorato, M., Librino, F., Zorzi, M.: Integrated Cooperative Opportunistic Packet Forwarding and Distributed Error Control in MIMO Ad Hoc Networks. IEEE Transactions on Communications 59(8), 2215–2227 (2011)
15. Zeng, K., Lou, W., Zhai, H.: On End-to-End Throughput of Opportunistic Routing in Multirate and Multihop Wireless Sensor Networks. In: Proceedings of INFOCOMM (2008)
16. Wang, S., Cheng, S.: Opportunistic Routing in Intermittently Connected Mobile P2P Networks. IEEE Journal on Selected Areas in Communication 31, 369–379 (2013)
17. Shin, W.-Y., Chung, S.-Y., Lee, Y.H.: Parallel Opportunistic Routing in Wireless Networks. IEEE Transactions on Information Theory 59(10), 6290–6391 (2013)
18. Passarella, A., Kumar, M., Conti, M., Borgia, E.: Minimum-Delay Service Provisioning in Opportunistic Networks. IEEE Transactions on Parallel and Distributed Systems 22(8), 1267–1275 (2011)
19. Myung, J., Lee, W.: Wireless Sensor Networks: A Survey. IEEE Communication Letters 16(4), 510–513 (2012)

A Security Framework for Multi-authority Based e-Sealed Bid Auction System

Sourav Sinha and Mamata Jenamani

Department of Industrial Engineering and Management, Indian Institute of Technology,
Kharagpur, India
sourav.sinha@iitkgp.ac.in, mj@iem.iitkgp.ernet.in

Abstract. In a typical organization multiple authorities may participate in purchasing process. Online implementation of such system should properly distribute security control over multiple-authorities to ensure trust among them and avoid dispute. In our research we are presenting two aspects of e-sealed bid auction in a two server setting using public key infrastructure: First, secure bid submission and bid opening using multiple authorities, and second, secure bid document storage for future references.

Keywords: Sealed bid auction, Multi-authority auction, E-tendering, Bid encryption decryption.

1 Introduction

Government organizations follow tendering process for procuring products or services. In this process a set of suppliers submit bids in sealed envelope. Online implementation of this process is called e-sealed bid auction system [6]. Such systems significantly reduce the paper work and increase the transparency in the system [18], hence, strongly recommended by Government of India along with other e-procurement activities [16].

In a typical e-sealed bid auction system security is a major issue as the bid values, submitted online and stored in a server, is to be revealed on a specific date and a specific Evaluation Committee (EC). The issues become more complicated in a case of a two cover bidding system that is generally preferred by the government organizations. A two cover system comprises of a *techno-commercial bid* and a *price bid*. At the time of bid opening, first the techno-commercial bids of the all bidders are opened and evaluated by the Evaluation Committee Members. Only those quotations qualify to the second round which matches with the technical specification set up by the buyer. In the second phase the price bids are opened. This process increases the vulnerability of the bids being exposed to adversaries. Therefore, a well-structured e-sealed bid auction system should protect the common interest of the bidders as well as the buying organization from collusion with peers as well as with auctioneer. Presence of various members in different capacities in the evaluation committee both in departmental and central purchasing process under a distributed system setting makes security issues even more complex. The major security issues or threats identified in the literature [10, 21, 22, 24] in online auction implementation are listed

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 32–44, 2014.

in Table 1. As shown in the table the security issue can be broadly classified into seven areas.

1. *Confidentiality of bid values*: Bids values stored on the systems are to be protected against unintended or unauthorized access. Only the bidder should know their bid values until the closing period.
2. *Fairness & trust*: Clarity should be maintained in the overall process so that bidders can keep faith on it. No one can disclose the content of any of the bids until the bidding procedure is closed.
3. *Authenticity of the bidders*: Bidders must be authenticated by their digital certificates issued from a mutually trusted third party (CCA approved Certifying Authority). No malicious intruder should get into the system.
4. *Verifiability of bids*: All bids should be publicly verifiable. At the end of the auction, if any bidder challenges his bid against the winning bid, system should keep provision to verify it.
5. *Non-repudiation*: No bidders can deny the bids that they have categorically submitted. Once a bidder confirms his participation in bidding, he cannot deny any bid submitted against the items.
6. *Bidder's anonymity*: Bidders identity should not be made public even when bids are opened. Only the winning bidder's identity and his bid price can be published.
7. *Privacy of losing bids*: All bidding prices and bidders identity (except the winner) are not revealed to anyone.

Table 1. The major security issues identified by different researcher in implementing e- auction

Security Issues	Authors
Confidentiality of bid values	Naor, Pinkas, and Sumner 1999;Viswanathan, Boyd, and Dawson 2000; Liao,Wang and Tserng 2002
Fairness & trust	Naor, Pinkas, and Sumner 1999;Liao,Wang and Tserng 2002; Shih, Yen, Cheng and Shih 2011
Authenticity of the bidders	Naor, Pinkas, and Sumner 1999;Juels and Szydlo 2003; Shih, Yen, Cheng and Shih 2011; Yang, Nasemri, and Wen 2009;Xiong, Chen, and Li 2013
Verifiability of bids	Juels and Szydlo 2003; Shih, Yen, Cheng and Shih 2011;Liao,Wang and Tserng 2002
Non-repudiation	Viswanathan, Boyd, andDawson 2000;Juels and Szydlo 2003, Shih, Yen, Cheng and Shih 2011
Bidder's anonymity	Viswanathan, Boyd, and Dawson 2000; Yang, Naseri, and Wen 2009; Li, Juan, and Tsai 2011
Privacy of losing bids	Brandt 2000;Xiong, Shih, Yen, Cheng and Shih 2011;Xiong, Chen, and Li 2013

Besides the above security issues, an auction system in typical organization may face *security control distribution problem* among multiple authorities and we are probably the first to investigate this issue in the context of public procurement. In this paper we propose an online implementation framework traditional tendering system considering a multiple authorities. In particular we extend the work presented in Naor et al. [13] and Juels and Szydlo [7] of two-server based sealed bid auction system. However, we use public key infrastructure instead of garbled circuit [27] in line with the work of Seveda [19]. While [22] uses bidder's public key for encryption purpose, we use the public keys of a group of stakeholders who act as the members of some purchase committee using multi-authority based encryption technique. The proposed framework also includes a secure bid document storing strategy [19] for future reference. Thus, we contribute to the body of literature by proposing a secure sealed bid auction system that takes care of bid submission, opening and document storage using multiple authorities.

The rest of the paper is organized as follows: *Section 2* represents literature review, which is categorized in five groups. *Section 3* studies the traditional tendering system of a typical government organization and find out the lapses in such system. At the same time, in this section we have restructured the traditional system to overcome the identified problems and fit it in electronic auction system. *Section 4* describes the main entities, overall architecture and the implementation phases of the proposed e-sealed bid auction framework.

2 Literature Review

There exists three approaches for designing secure sealed bid auction system using: 1)*Garbled Circuit* [27, 7, 9, 13, 30], 2)*Public Key Infrastructure* [2,19,22], and 3) *Quantum Key Distribution* [37, 14, 26, 36]. Naor et al. [13] are first to propose a sealed bid auction protocol based on two-server system. It is the most popular sealed bid auction protocol (NPS Protocol) of that time [7]. The main entities of this protocol are Auctioneer, Auction-issuer and the Bidders. An auctioneer is a third party, who holds the auction. Auction-issuer is a service provider and the bidders are the vendors. This protocol uses garbled circuit [27] for encrypting bid values given by the bidders. If we ignore the possibility of collusion among any party, this protocol works well. But, if any collusion occurs, there is no way to trace the cheating and it appears almost clueless. Juels and Szydlo[7] try to overcome these drawbacks in a two server setting. Later, Hinkelmann et al.[9]propose another protocol where they confirm no information leak up to t-party collusion. Here, again the main concept of bid encryption is based on garbled circuit. The main disadvantage of using garbled circuit is its computational complexity [8].It takes considerably large memory space for computation. Kudo [29] propose electronic sealed bid auction in public key infrastructure where they introduced a time-key concept. This protocol is three servers based; auction server, key registering server and time server. Viswanathan et al. [22] propose a sealed bid auction schema where they segregate complete process in three phases. In the initial phase they reused Franklin and Reiter‘s [32] verifiable signature sharing technique to achieve bidders' anonymity. The other two phases of this protocol are very common; bidder registration phase and bid submission phase. A European patent [19] implement a security framework of sealed bid auction based on public key infrastructure. Though the proposed framework is basically the generalized concept of document management system, here they mention that this works fine for electronically sealed bidding system as well. Naseri [14] first introduce use of quantum key distribution in sealed bid auction system. This paper is extended by Yang et al. [26] to improve auction security.

Seale bid auction may not confine in single server. External entities are often used as a service providers provider [7, 9, 1319, 22]. Secure communication among these servers is an important issue in any such multi-server based system. A two server setting first proposed by Naor et al.[19] uses a Proxy Oblivious Transfer (POT) Protocol to achieve secure communication between two servers. POT protocol is originally the extension of 1-out-of-2 oblivious transfer [38]. In this protocol, the bidder simply acts as a chooser, auction issuer act as a sender and the auctioneer act as a proxy. Juels and Szydlo (2003) modify this concept to impose bid-verifiability. But, both the protocols do not incorporate publicly verifiability of bids. With the advent of Quantum Key Distribution (QKD) [37], sealed bid auctions get a new direction. Naseri[14] used Quantum Key Distribution for secure communication. This protocol further extended by Yang et al. [26] to improve its performance. Brandt [2] and Li et al. [10] use Diffie–Hellman key exchange algorithm for communication among different entities. Currently, web services are used to provide secure communication for distributed computations [17, 39, 34].

A bidding system should ensure proper sealing of bids. Different encryption techniques are used for sealing the bid values. A number of literatures rely on Yao's [27] garbled circuit for this purpose [7,9,13]. Garbled circuit uses binary XOR gates for encrypting the values. Bid values are straightway encrypted and stored in auctioneer server. But, computational complexity [8] is very high in any garbled circuit and there is no provision for bid value archiving in these systems. On the other hand, use of public key infrastructure in sealing bid value can be classified in two different approaches based on the encryption techniques used [1].In first approach, bidder himself open the bid [22] at the time of bid opening. In this case bidder's public key may directly be used in sealing bid values [19]. But, this is not a good idea because it makes bidders' presence (online) necessary at the time of bids opening [1]. In second approach, auctioneer becomes the bid opening authority [31, 32] and this technique is mostly used in real implementation. In recent time quantum mechanics is being used widely [14, 26]. But, this technology is not popularly used in public buying, probably because of its high installation cost. For example, Seveda[19] proposed bid document encryption using public key infrastructure. This framework provides a secure way to store bids in an encrypted document format. Olive [15] proposed secure storing and archiving technology based on public key infrastructure in a two server setting. Though, it is not a part of a sealed bid auction framework but the concept of secure document storing can be adopted in e-auction.

Multi-authority based encryption are necessary for distribute security control among a number of authorities. Chase et al. propose encryption schemes in which any number of independent authorities can be allowed to be involved in attribute based encryption (ABE) [3,4]. In a multiparty computation homomorphic key generation technique is used as described by Cramer [44].This homomorphic key is used for multiparty encryption-decryption operations. Lv et al. [12] propose message encryption protocol using group key for secure group communication in dynamic peer system. Zhou et al. [28] proposed a role based access control protocol over multiple authorities [28]. Xiong et al. extend Identity-based encryption protocol [20] to a multiple authority based encryption scenario [25].

Sealed bid auctions are online implementation of the traditional tendering system. Hence, it is necessary to have a clear understanding of it. A number of literatures reviewed with the interest of understanding the traditional tendering system and organizational purchase workflows. For example, Liao et al. [11] describe tendering process used in Taiwan Government. Panayiotou et al. [41] have shown traditional purchase procedure of the Greece Government. They discuss the conversion process to be followed for moving traditional system to B2B e-procurement setting. Du [33] develops an electronic platform B2B e-tendering model for representing traditional tendering system. Another author Du et al. [35] represent a study on the legal obligations in integrating a complex tendering system and at the same time also provide its implication on security aspects of an online tendering system.

3 Study of Work-Flow in Traditional Tendering Process

In a Government organization most purchases are conducted centrally. Sometimes, provisions are kept at department level for limited and small amount purchases mainly for emergency purposes. Here, the signing authority is the department head. But when the amount exceeds certain limit, central committee takes over. In this situation, the purchase requisitions are raised from the departments and sent to the competent authority for approval. An approved purchase proposal comes to the central purchase section for further proceedings. If similar proposals from many departments come in between, all are clubbed together and jointly placed to the auction authority for further processing. Now, it is the prerogatives of the said authority to arrange an auction, release tender, interact with vendors, take bids and select winner in a fair-play. Depending on the outcomes of the auction, buyer awards the order to the potential company and request to dispatch the materials on or before the deadline.

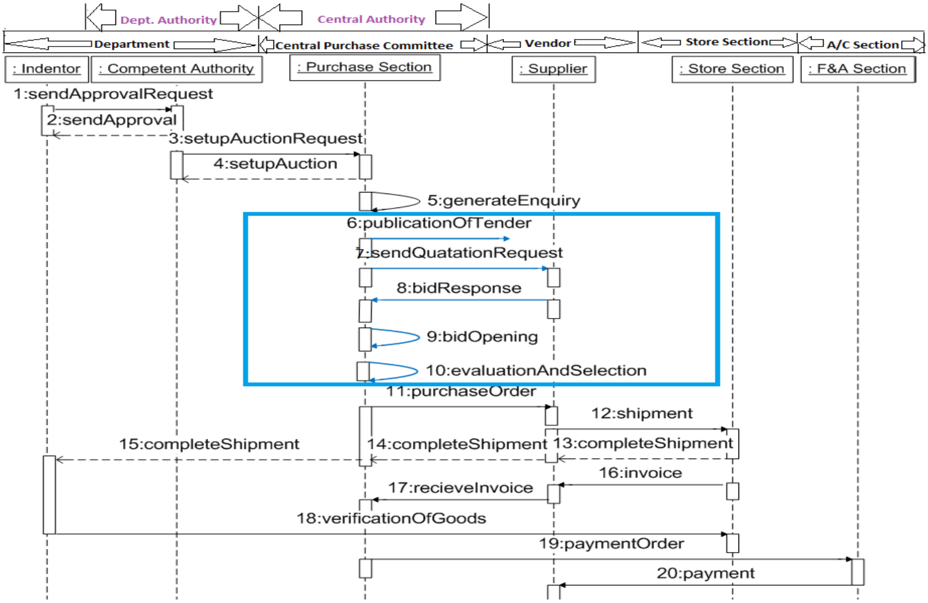


Fig. 1. Work flow of a traditional tendering process of a typical organization

Work-flow of the traditional tendering process of a typical government organization is shown in Fig.1using a sequence diagram. As shown in the figure, most of the major activities (marked area in Fig. 1) are directly controlled by the central committee comprising members with different capacities. It is desirable that, the security control needs to be distributed among the committee members to increase the level of trust and decreasing the chance of collusion. The proposed online implementation ensures participation of the all the authorities (committee members). We limit our work to the activities included in the blue rectangle in Fig.1. It mainly includes the bid submission, storage and opening stages in a two cover system.

In the corresponding online implementation the major concern is proper sealing of bids. This requires a number of additional activities that are not the part of traditional system. Fig. 2 represents workflow of the proposed electronic bidding system with distributed security control over multiple authorities. In this system, at the time of enquiry generation an Evaluation Committee is formed for governing the bid evaluation process. Sealing of each bid-document is done by the public key of any two Members of the Evaluation Committee (say member i and j) randomly chosen by the system. In case of multi authority based purchase, these committee members comprise Central and Departmental purchase committee representatives. Some constraints may be imposed to such system so that members with varied interest groups are uniformly presented. For example, if member i represent central committee, member j may be selected from departmental committee. Here, an effort has been made to distribute the control of activities over multiple authorities while making a proper trade off with the computational time.

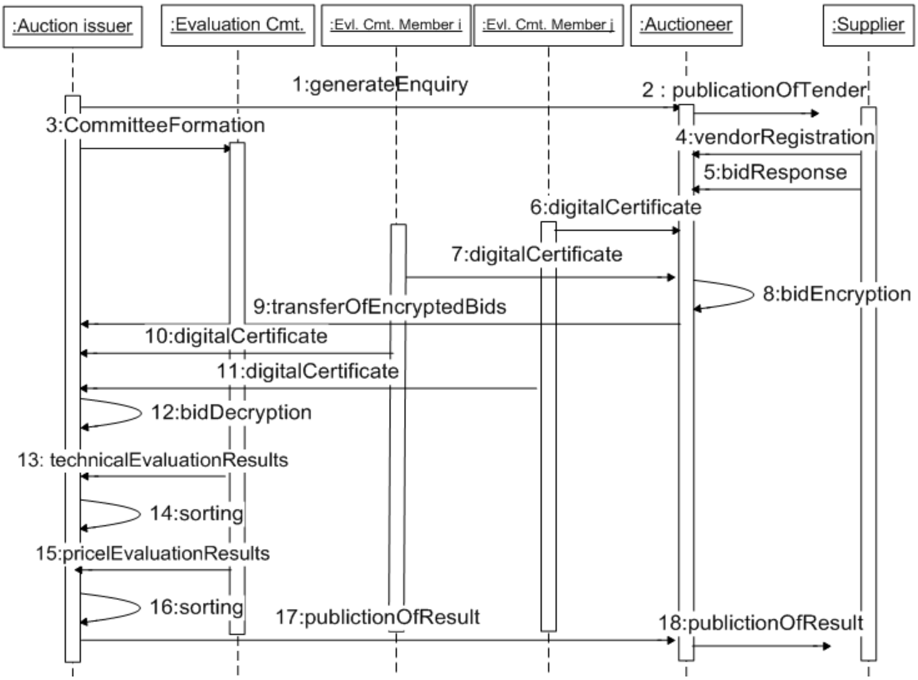


Fig. 2. Portion of the e-tendering system that has been redesigned

4 Proposed e-Sealed Bid Auction Framework

The proposed framework requires two servers. Four entities interact during the secure operations. The framework requires that all the entities are part of some public key infrastructure. As described below:

4.1 Main Entities of the Proposed e-Sealed Bid Auction Framework

This framework connects four entities. The brief introductions of these entities are as follows;

1. Bidder (Bi): The vendor company(s) that take part in the bidding. They submit their bids to the auctioneer server. At the end of the auction bidders comes to know about the outcomes of the bidding from the auctioneer.
2. Auctioneer (A): A third party who runs the auction. Auctioneer acts as a service provider. It advertises the auction, take bids from bidders and transfer the bids to the auction-issuer for bid evaluation and winner selection and ultimately publish the outcomes of the auction.
3. Auction-issuer (AI): The buyer. He hires/appoint the auctioneer for holding the auction and auctioneer transfer the bids to auction-issuer for further processing.
4. Evaluation Committee Members (ECMs): The Purchase Committee members of buyer side. Bid encryption keys are encrypted with the public key of the Evaluation Committee Members (ECMs). Bids can be opened only with the ECMs private keys.

4.2 Implementation Phases of Proposed Framework

Auction-issuer hosts an internal server. Access control of this server restricted within the buyer organization. Its only connection to the outside world is with the auctioneer server through some secure web-service. A strong firewall needs to be installed between auction-issuer and outside world with such a configured that only auctioneer can get into the system with some restricted privilege [5]. Thus auction-issuer has been kept abstract to the vendor level. The sequence of phases that to be carried out by the proposed framework are: 1) Online vendor registration, 2) Bid submission by authorized vendors, 3) Secure framework for bid encrypting and storing and 4) Bid Decryption/opening. Fig. 3 is the diagrammatic representation of the proposed framework. The security issues that make the system vulnerable to various types of attacks in phase of framework are identified and briefly examined in each section. We now discuss the approach to be followed in each of these activities and the security analysis:

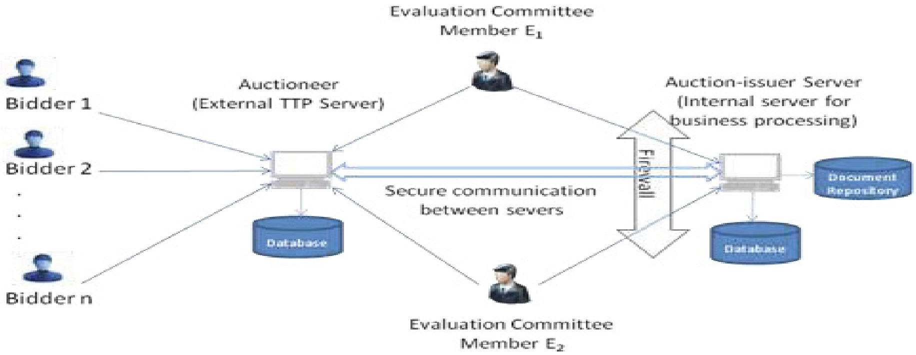

Evaluation Committee Member E_1
Bidder 1
Bidder 2
Bidder n
Auctioneer (External TTP Server)
Auction-issuer Server (Internal server for business processing)
Firewall
Secure communication between severs
Database
Document Repository
Database
Evaluation Committee Member E_2

Fig. 3. Proposed e-Sealed Bid Auction framework

4.2.1 Online Vendor Registration

Vendors those who opt for e-auction, must register themselves in the auctioneer server prior to bidding. The vendor has to use a CCA certified digital certificate issued by mutually trusted certifying authority to ensure authenticity. Vendor Registration may consist of the following phases;

1. Get Vendor Details: Vendor first enters the information about company, contact detail, contact person details, product/service details. Most importantly verification and validity checking of digital certificate is done by the system, at this time.

2. Verify Past Records: Before coming to a big contract or awarding an order to a supplier/service-provider it is important to verify the past recodes and potential to deliver the order. Candidature of the company may not be considered if the past record is not good or it is black listed.

3. Issue Login Credentials for Authentication: At the end phase an account is created in the name of the vendor, along with user login-id and password. They can update given information from time to time and get a view of all the available tender for which they are eligible.

Security Analysis

*Authenticity of the bidder***.** Digital certificate (version 3) issued by any of the CCA certified Certifying Authority is widely recognized and trusted for authentication. A new vendor must possess a digital certificate (X.509 format) at the time of vendor registration. Some of the essential information (vendor name, public key of the certificate etc.) is directly captured from the certificate at the time of registration by an internal applet program call. This program also checks the validity and authenticity of the certificate.

Confidentiality of bidder. A register vendor owns a secure user account where all information is kept private. An authenticate vendor can login to the system using his login credential. Standard mechanism is used for securing user account. Here a user may hold multiple stakes so we use Single Sign-On (SSO) Token for secure login. We hold the token in the session. For storing the password of the user securely hashing algorithm (MD5) has been used.

4.2.2 Bid Submission by Authorized Vendor

Authorized vendor can login to the system with their login credentials and can take part in bidding against any of the available tenders at that time. All data between auctioneer and bidder is exchanged using SSL protocol. A vendor take part in a fresh bidding a unique ID is generated. We call this ID as Bid ID in rest of the paper. Bid is registered against this ID, not against vendor name. To achieve bidders' anonymity we never store bid ID of any bidder in the system against vendor name. At the time of bid submission two different documents are generated simultaneously. First one is generated at client (bidder) side, blindly signed by auctioneer. This document contains all bid values submitted by the bidder and the system generated unique bid ID of the bidder. A bidder can use this document as a legal document to solve any dispute, if occurs. Second one is an encrypted XML file generated at server (auctioneer) side. This encrypted bid document is tagged against the same unique bid ID not by vendor name. The detail of the bid encryption and storing is discussed in next phase of bidding.

Security Analysis

Bidder's anonymity. Bidder's identity is never recorded against the bid document submitted by a bidder. Rather, a system generated random ID is used for tagging the bid document. At the end of bid opening phase comparative statement is also generated against this bid ID. Post evaluation the winner's bid ID is made public so that actual bidder can claim the award.

Fairness and Trust. The basic security enforcement use in proposed framework based on public key infrastructure and this is implemented using digital certificate issued by CCA certified certifying authority. CCA is widely trusted and it is appointed by the Ministry of Communication and Information Technology, Government of India. In this system bids are registered against a unique ID and throughout the process bidders' anonymity is maintained so that no bidder gets any favour. Integrity of bid values is also preserved, only bidder can modify his bid till last date. No other can modify or see the bid.

4.2.3 Secure Framework for Bid Encrypting and Storing

In the following section we are proposing a bid encryption and storing strategies:

1. Bid Document Encryption: Bidder B_i submits bid on-or-before the dead line. System will generate a random key R_0 to encrypt the bid-document b_i. We are considering that auctioneer system is only permitted to store the encrypted bid document; they cannot store the original document before encryption. Once the document is encrypted with the secret key R_0, document is safe until any one get access to the key.

$$E(b_i , R_0)$$

Now, the concern is to hide the key properly so that no one gets access of it.

2. Serial Encryption of Bid Document Encryption Key (R_0): System can generate another two random key (R_1 and R_2) for serially encrypting the first encryption key R_0. Here we are using 3DES [23] with two keys for serial encryption of R_0. First encrypt R_0 with R_1 using symmetric key encryption. The output of this encryption is fed as an input to the second encryption and encrypted with R_2. Finally, again encrypt the output with R_1 and store the encrypted output in the database.

$$3DES_{R1,R2,R1}(R_0)=C_{R12}$$

3. Selection of Evaluation Committee Members: In this phase our aim is to hide the 3DES encryption keys (R_1andR_2) so that no one can recover R_0 even if get able to access the encrypted key C_{R12}, is stored in database. We are proposing here to use public key of the Evaluation Committee Members (ECMs) to encrypt these keys. ECMs are divided in two groups, first group representing Departmental Purchase Committee (DPC) and other representing Central Purchase Committee (CPC), as stated earlier. One member from each group is selected randomly by the system. Identity of the person is not revealed. Say, E_1 is chosen from the Departmental Purchase Committee and E_2 from the Purchase Committee to encrypt the bids of the bidder B_i.

4. Encryption of 3DES keys ($R_1$and R_2): The 3DES keys ($R_1$and R_2) are encrypted with public key of the Evaluation Committee Members ($E_1$and E_2) and the encrypted keys are stored in the database. Here Pub_{E1} and Pub_{E2} represent the public keys of E_1 and E_2 respectively.

$$E(Pub_{E1}, R_1) = E_1R_1$$
$$E(Pub_{E2}, R_2) = E_2R_2$$

Finally these encrypted keys (E_1R_1, E_2R_2) along with the encrypted bid-document are transferred to the auction-issuer server through some secure web-service for bid opening phase. The encrypted bid document is stored in document repository at the auction issuer side and the encrypted keys are stored in the database. These bid document and encrypted keys are stored against the bid ID not against the vendor name.

Security Analysis

Non-repudiation. An authenticate bidder can participate in bidding process. Use of digital certificate restricts the scope of false bidding. Once bid is submitted vendor cannot deny it and others cannot alter it. Only those bid documents submitted properly by the bidder before the deadline are considered as valid bid.

Confidentiality of bid values. Bid values are never stored before the encryption operation is done. At the time of final submission, an encrypted bid document is generated and transferred to the repository for secure storage. For proper encryption of bid document we use three levels encryption process as described in the bid encryption phase. Keys are also encrypted in such a fashion that even the system administrator/ database administrator cannot decrypt the bid document using these keys.

4.2.4 Bid Opening

Bid opening is the decryption process. All Evaluation Committee Members of the both committees (Departmental and Central) need to be present online.

1. Recovering 3DES keys (R_1 and R_2). On a pre-specified time system starts checking all possible combination among Departmental ECMs and Central ECMs to find required match to open the bids. Decryption process follows the same sequence as encryption process but in reverse order. E_1 supplies his private key (PR_{E1}) first and recovers R_1 recovered. Same way R_1 can be obtained when E_1's his private key (PR_{E2}).

$$D(PR_{E1,} E_1R_1) = R_1$$
$$D(PR_{E2,} E_2R_2) = R_2$$

2. Recovering document encryption key (R_0). Once R_1 and R_2 are obtained, we serially decrypt C_{R12} to recover R_0. The sequence of 3DES operations comes as Decryption-Encryption-Decryption, just reverse of the encryption order. At the end of the full operation 3DES serially decrypt the encrypted keys and R_0 is recovered.

$$3DES_{R1,R2,R1}(C_{R12})=R_0$$

3. Document Decryption. Once the original secret key R_0 is obtained the bid-document can be decrypted using this key easily.

$$D(E(b_i , R_0) , R_0)= b_i$$

For all encrypted bid-document same set of operations are followed to recover the original document. This original document can be stored in the repository for future reference. Finally, when the bid-opening phase is over for all bidders, the comparative statement is generated against the bid ID and it is placed to the evaluation committee for further processing. Evaluation committee selects the winner (or group of winner) and intimate auctioneer about the winner's bid ID. Auctioneer communicates to all the bidders so that winner can claim the award with the valid bid document generated at bidder side and signed by the auctioneer at the time of bid submission.

Security Analysis

Privacy of losing bidder. At the end of bidding process auctioneer only reveals the winning bid ID. No name is formally announced by any official. Even no detail of the participants of the bidding is also announced. Full privacy of the losing bidders and bid values are preserved.

Verifiability of bids. If any dispute arise post declaration of winning bid ID by auctioneer, any bidder may appeal against the decision. Bidder may claim directly to the auction issuer within a time frame with the bidder's document blindly signed by the auctioneer. This document is a legitimate proof in favour of the bidder. If any discrepancy noticed in the bid values present in auction issuer server and bidder's document, necessary steps can be taken.

5 Conclusion

The proposed security framework for e-sealed bid auction efficiently addressed bid submission, opening and secure document storage using multiple authorities. This framework also defended the security control distribution problem that was identified in a multi authority based purchasing system and we are probably the first to investigate this issue in the context of public procurement. As the objective of this paper is security framework design, our focus was not the strategic aspects of bid evaluation or other decision making criteria for winner selection. Another serious limitation that can be noted here is multi auctioneer or multi auction-issuer setting does not fit here. In practice buyer may prefer multi auctioneer model to enlarge the scope of getting desirable seller.

References

1. Boyd, C., Mao, W.: Security issues for electronic auctions. Hewlett-Packard Laboratories, HPL-2000-90 (2000)
2. Brandt, F.: How to obtain full privacy in auctions. International Journal of Information Security 5(4), 201–216 (2006)
3. Chase, M.: Multi-Authority Attribute Based Encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007)
4. Chase, M., Chow, S.M.: Anonymous key issuing for Authority Attribute Based Encryption, US Patent,12(355),862 (2007)
5. Girouard, J., M., Ratliff, E., J., Simon, K.D.:System and method for intrusion decision-making in autonomic computing environment, US Patent, US 2005 0278178 A1(2005)
6. Islam, M.S., Dey, S., Kundu, G., Hoque, A.S.M.: A solution to the security issue of an e-government procurement system. In: International Conference on Electrical and Computer Engineering, pp. 659–664 (2008)
7. Juels, A., Szydlo, M.: A Two-Server, Sealed-Bid Auction Protocol. In: Blaze, M. (ed.) FC 2002. LNCS, vol. 2357, pp. 72–86. Springer, Heidelberg (2003)
8. Henecka, W., Schneider, T.: Two-Party Computation with Less Memory. In: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, pp. 437–446 (2013)
9. Hinkelmann, M., Jakoby, A., Stechert, P.: t-Ptivate t-Secure auction. Journal of Computer Science and Technology 23(5), 694–710 (2011)
10. Li, M., Juan, J., Tsai, J.S.T., Practical, J.H.C.: electronic auction scheme with strong anonymity. Information Science 181, 2576 (2011)
11. Liao, T., Wang, S., Tserng, M.T., Framework, H.P.A.: of electronic tendering for government procurement: a lesson learned in Taiwan. Automation in Construction 11, 731–742 (2002)
12. Lv, X., Li, H.S., Wang, B.: Group key agreement for secure group communication in dynamic peer systems. Journal of Parallel and Distributed Computing 72(10) (2012)
13. Naor, M., Pinkas, B., Summer, R.: Privacy preserving auctions and mechanism design. In: ACM Conference on Electronic Commerce, vol. 1, pp. 129–139 (1999)
14. Naseri, M.: Secure quantum sealed-bid auction. Optics Communications 282, 1939–1943 (2009)
15. Olive, J.: Secure Document Management system, US patent, US 2008 0235175 A1 (2008)
16. Public Procurement Bill: Planning Commission, Government of India (2012)
17. Pei, S., Chen, D., Chu, Y., Xu, Q., Xi, S.: Research of web service security model based on SOAP information. Information Technology Journal 11(2), 241–247 (2012)
18. Richhariya, P., Singh, P.K.: A Survey on Financial Fraud Detection Methodologies. International Journal of Computer Applications, 45–22 (2012)
19. Seveda, R.: Document Security Management system, EP patent, EP 1 894866 B1 (2011)
20. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
21. Shih, D.H., Yen, D.C., Cheng, C.H., Shih, M.H.: A secure multi-item e-auction mechanism with bid privacy. Computer & Security, 273–287 (2011)
22. Viswanathan, K., Boyd, C., Dawson, E.: A Three Phased Schema for Sealed Bid Auction System Design. In: Clark, A., Boyd, C., Dawson, E.P. (eds.) ACISP 2000. LNCS, vol. 1841, pp. 412–426. Springer, Heidelberg (2000)
23. William, B.C., Elaine, B.: Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, NIST Special Publication, 800-67 (2012)

24. Xiong, H., Chen, Z., Li, F.: Bidder-anonymous English auction protocol based on revocable ring signature. Expert Systems with Applications, 7062–7066 (2013)
25. Xiong, H., Yuen, T.H., Zhang, C., He, Y.-J., Yiu, S.M.: Attribute specified identity-based encryption. In: Deng, R.H., Feng, T. (eds.) ISPEC 2013. LNCS, vol. 7863, pp. 60–74. Springer, Heidelberg (2013)
26. Yang, Y.G., Naseri, M., Wen, Q.Y.: Improved secure quantum sealed-bid auction. Optics Communications 282(20), 4167–4170 (2009)
27. Yao, A.C.: Protocols for secure computations. In: FOCS 1982, pp. 160–164. IEEE Computer Society (1982)
28. Zhou, L., Varadharajan, V., Hitchens, M.: Enforcing Role-Based Access Control for Secure Data Storage in the Cloud. The Computer Journal 54(10), 1675–1687 (2011)
29. Kudo, M.: Secure electronic sealed-bid auction protocol with public key cryptography. IEICE Trans. Fundamentals E81-A(1), 20–27 (1998)
30. Kolesnikov, V., Sadeghi, A.-R., Schneider, T.: Improved garbled circuit building blocks and applications to auctions and computing minima. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 1–20. Springer, Heidelberg (2009)
31. Sako, K.: An auction protocol which hides bids of losers. In: Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS, vol. 1751, pp. 422–432. Springer, Heidelberg (2000)
32. Franklin, K.M., Reiter, K.M.: The design and implementation of a secure auction service. IEEE Transaction on Software Engineering 22(5), 302–312 (1996)
33. Du, T.C.: Building an Automatic e-Tendering System on the Semantic Web. Decision Support Systems 14(1), 13–21 (2009)
34. Benatallah, B., Casati, F., Toumani, F.: Web Service Conversation Modeling: Cornerstone for e-Business Automation. IEEE Internet Computing 7(6) (2003)
35. Du, R., Foo, E., Boyd, C., Fitzgerald, B.: Defining security services for electronic tendering. In: The Australasian Information Security Workshop (AISW 2004), vol. 32, pp. 43–52. Australian Computer Society Inc. and ACM (2004)
36. Zhao, Z., Naseri, M., Zheng, Y.: Secure quantum sealed-bid auction with post confirmation. Opt. Commun. 283(16), 3194–3197 (2010)
37. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing, Bangalore, India, pp. 175–179 (December 1984)
38. Beaver, D., Micali, S., Rogaway, P.: The round plexity of secure protocols. ACM Synopsis on Theory of Computing, 503–513 (1990)
39. Kabisch, S., Peintner, D., Heuer, J.: XML-based Web service generation for microcontroller-based sensor actor networks. In: 2010 8th IEEE International Workshop on Factory Communication Systems (WFCS), pp. 181–184 (2010)
40. Cramer, R., Damgård, I., Nielsen, J.B.: Multiparty computation from threshold homomorphic encryption, pp. 280–300. Springer, Heidelberg (2001)
41. Panayiotou, N.A., Sotiris, P.G., Ilias, P.T.: An e-procurement system for governmental purchasing. International Journal of Production Economics 90(1), 79–102 (2004)

Analysis of Semantic Attacks in Online Social Networks

K.P. Krishna Kumar and G. Geethakumari

BITS-Pilani, Hyderabad Campus, Hyderabad, India
kpkrishnakumar@gmail.com,
geetha@hyderabad.bits-pilani.ac.in

Abstract. The emergence of online social networks as an important media for communication and information dissemination during the last decade has also seen the increase in abuse of the media to spread misinformation, disinformation and propaganda. Detecting the types of semantic attacks possible in online social networks would require their accurate classification. Drawing similarities with other social computing systems like Recommender systems, this paper proposes a new taxonomy for semantic attacks in social networks. Further, we propose an algorithm which uses social network as a medium for social computing to analyse the patterns of propagation of information and identify sources of misinformation in them. We construct a new information propagation graph from the social network data and carry out k-core decomposition of the graph to isolate possible contents of misinformation and the user nodes which are involved in their propagation. We used seven different data sets obtained from 'Twitter' to validate our results.

Keywords: online social network, semantic attacks, social computing, disinformation, misinformation.

1 Introduction

There has been an exponential proliferation of Internet and the emergence of Web 2.0 technologies in the last decade. Online Social Networks (OSNs) have become important media for communication as well as information dissemination. The availability of multitude of mobile platforms like smart phones, tablets, cameras apart from traditional personal computers and laptops to assess Internet has made social networks an ubiquitous source of information. However, lack of accountability and traditional censorship in OSNs have increased the spread of misinformation in social networks.

OSNs are also social computing platforms much like the other such systems viz Recommender systems and Reputation systems. Recommender systems are used to recommend items to other users based on similar ratings or similarity in contents. Reputation systems are used to establish trust relationships based on the ratings given by different users in the network. Similar to these systems, the information contents of news items in OSNs could also be peer reviewed for their quality and could be used to identify potential misinformation and their sources. Like any social computing platforms, OSNs are also not free from manipulation. Manipulation of social network users to accept false information and causing its spread in the network comes in the realm of semantic attacks. The main contributions of this paper are:-

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 45–56, 2014.

– Propose an accurate taxonomy of semantic attacks in OSNs based on extensive study of real world data sets.
– Develop OSNs as platforms of social computing to detect spread of misinformation.
– Propose a computationally efficient algorithm to identify sources of misinformation for their monitoring.

The rest of the paper is organised as follows. We give the latest developments in the field and also discuss semantic attacks in other social computing systems in Section 2. We discuss semantic attacks in OSNs and propose a taxonomy for their analysis using different data sets in Section 3. In Section 4, we propose an algorithm to detect semantic attacks by colluding users and in Section 5 we conclude by outlining future direction of work in this area.

2 Related Work

Information diffusion in OSNs like Twitter, Facebook, LinkedIn etc have been studied extensively in the literature. Twitter is a micro blogging site which requires the users to post messages or tweets in a maximum of 140 characters. Calamities like earth quakes and other important events like the 2011 Egyptian revolution were extensively covered in Twitter [12] [15]. It may even be appropriate to say that news no longer breaks, it tweets [13]. The factors which govern the influence of news items in social networks are indegree, page rank, retweets, mentions, influence trees etc [3]. The information propagation and influence mechanism in large scale Twitter networks could be studied using the 'retweet' feature in Twitter [9]. Supervised learning techniques to detect suspicious memes in microblog platforms like Twitter has been done in [11]. All of these techniques use machine learning algorithms or heuristics to detect misinformation contents in the posts made in the OSN. Our approach differ from these in that we reverse the process of content analysis of the messages and their possible spread in the network. We carry out network analysis of information propagation in OSNs to segregate possible misinformation. This reduces the computationally intensive semantic analysis to only the filtered data to verify the information contents.

Recommender systems are important social computing systems. The openness of Recommender systems are a source of strength and vulnerability. The ratings of items in Recommender systems are manipulated by *Shill attacks* [1]. Shill attacks are a result of rating items of preferred sellers so as to push up their ratings called *push attacks* or pull down the ratings of other sellers called *nuke attacks*. Shill attacks - both push attacks and nuke attacks are carried out by the introduction of biased user recommendation profiles in the system. This could be in the form of sybil attacks, where multiple identities are created by a single user or entity. The different types of shill attacks could include Random Attack, Average Attack [5], Bandwagon or Popular Attack [10], Probe Attack, Segment Attack, Love/hate Attack [16], Sampling attack, Perfect Knowledge Attack [17] etc. These methods differ in the manner by which the ratings are given by the attackers to a profile of items so as to manipulate the recommendations of the system. We intend to use the methodology of the attackers in a similar manner to classify attacks of manipulation of information flow in OSNs.

3 Semantic Attacks in Online Social Networks

In this section we would explore the experiments conducted in a number of data sets to study the type of semantic attacks. Semantic attacks are aimed at influencing the perceptions of users with the aim to modify their actions. This would happen in OSNs through the spread of misinformation, disinformation, rumours, propaganda or lies.

The deliberate spread of false information is *disinformation*, which differs from *misinformation* only in terms of intent of the user. Disinformation is defined as 'deliberate falsehood', whereas misinformation is defined as 'accidental falsehood' [14]. A coordinated group of individuals, effectively using the underlying concerns of users of social networks, can cause semantic attacks.

Accuracy of the information is one of the important measures of quality of information. Honest mistake in the spread of inaccurate information is *misinformation*, whereas when the intention is to deceive the recipient, it is *disinformation* [6] [8]. While information, misinformation and disinformation are all informative in nature, only disinformation is deliberatively deceptive information and misinformation is misleading or false information.

3.1 Data Sets

We have done our experiments with data sets obtained from 'Twitter'. We give a brief description of the types of data sets we have obtained. Detection of misinformation is context specific and we obtained data sets from Twitter for different keywords to collect all tweets covering a specific period. The details of the data sets are given at Table 1. We used Twitter API to collect the tweets. The spreadsheet tool TAGS v5 used for collection of tweets using the Search API was provided by Martin Hawskey [7]. The data sets were obtained for events related to happenings in India and the World during the period from Jul to Oct 2013.

- **Egypt.** We investigated the spread of news related to the political unrest and massive protests in Egypt during the period from 13 Aug 2013 to 23 Sep 2013. The tweets were collected using the keyword 'egypt'.
- **Syria.** We tracked the events of use of chemical agents in Syria and all news related to it using the keyword 'syria'. The tweets were collected over the period between 25 Aug 2013 and 21 Sep 2013.
- **Bodhgaya.** The spread of information about terrorist attacks on 7 Jul 2013 at 'Bodhgaya' temple in India was tracked for a period of nineteen days from 07 Jul 2013 to 25 Jul 2013. The tweets were collected using the keyword 'bodhgaya'.
- **MyJihad.** We tracked a particular hashtag 'MyJihad' which we observed had contents which were controversial and the frequency of tweets were quite high. The tweets were collected over a period of eight days between 20 Jul 2013 and 27 Jul 2013.
- **Telangana.** The spread of politically sensitive information in India over the demand for a separate state of Telangana was studied using the keyword 'telangana'. The tweets were collected over a period of eight days between 23 Jul 2013 and 30 Jul 2013 prior to the government decision being announced.

- **Andhra.** There was wide spread stir against the bifurcation of the state of Andhra Pradesh in India after the decision was announced. We tracked the movement using the keyword 'andhra' and 'telangana'for the period from 30 Sep 2013 to 09 Oct 2013.
- **Phailin.** The coast of Odisha and Andhra Pradesh were hit by a severe cyclone 'Phailin' on 10-11 Oct 2013. We tracked the event in Twitter using the keyword 'phailin' for a period from 10 Oct 2013 to 13 Oct 2013.

Table 1. Details of data sets

Data set	Users	Tweets	Sources	Retweets	Period	Type
Egypt	27532	141682	10850	51723	13 Aug 2013 to 23 Sep 2013	Civil Movement
Syria	25415	104867	11452	44671	25 Aug 2013 to 21 Sep 2013	Political
Bodhgaya	4573	8457	660	4230	07 Jul 2013 to 25 Jul 2013	Terrorism
MyJihad	1166	5925	140	3232	20 Jul 2013 to 27 Jul 2013	Religious
Telangana	2671	6787	464	2177	23 Jul 2013 to 30 Jul 2013	Political
Andhra	3255	25463	1385	9064	30 Sep 2013 to 09 Oct 2013	Political
Phailin	4408	16190	1567	7408	10 Oct 2013 to 13 Oct 2013	Natural Calamity

3.2 Classification of Semantic Attacks

We carried out in depth analysis of the data using automated network analysis algorithms and human annotation of data, with the aim of understanding the types of semantic attacks possible in them. The data sets studied represent a wide domain and include events which attracted lots of comments in Twitter. On analysis, we found that around 20% in all the data sets relate to information which have no correlation with the event being discussed and the information in them could be classified as false information. There were around 30% of the information propagation in the data sets which are related to the event being discussed but making unsubstantiated claims and speculation.

We classify different types of semantic attacks based on the intention of the source of news items into two types:-

1. **Disinformation attacks.** We define Disinformation attacks as the spread of disinformation, propaganda and lies by a source knowing fully well that the information is false or misleading. The intention can be assessed by the efforts made in propagation of the news item and possibly the relevance of the information to the event being considered. In the analysis of the data sets we found disinformation being propagated in two forms - Sybil attacks and Shill Attacks.
   - *Sybil attacks.* When multiple identities are created in the social network to ensure the spread of a news item, the intention of the source is deliberate spread of false information. In the data sets, we saw that mostly such news items have little relevance to the event being discussed.

   - *Shill attacks.* As in Recommender systems, the users act as shills, collude with each other to cause the spread of a desired news item. The re-propagation of news item is construed as a positive affirmation about the acceptance of the quality of the item by the user and his intention to spread it.
2. **Misinformation attacks.** The intention of the source user is not to spread false information. This would include actual spread of biased opinions and false information which the sources believe to be genuine. Examples could also include feeds from news sites. Though collusion between users may not happen, the level of acceptance of such news items in the network would reveal patterns for their detection.

3.3 Construction of Retweet Graph

The collusion of users to promote or spread a certain news item of importance to the attackers and creation of multiple profiles as in Sybil attacks are two important methods which would be investigated. In a very large network like 'Twitter', it would be nearly impossible for a single user to spread misinformation to a large number of users. More often, it would require concentrated efforts by a group of users acting in consonance to achieve the aim.

The analysis of semantic attacks by a group of users acting in consonance to promote a single news item or a set of news items would be able to identified by increased communication between these users. In order to connect these users and the messages they propagate, we draw a 'retweet graph'. Retweet is a feature of Twitter which enables easy re-propagation of messages that one receives from others. We use this feature as a type of 'recommendation' of the news item and an endorsement of its quality by the user.

The retweet graph is constructed as a bipartite graph with two types of nodes. The aim is to group together news items initiated by the same sources and re-propagated by others. This would help us to understand collusion between sources as well as multiple profiles of the same user promoting same news item or similar news items. A retweet graph is constructed with social network users as well as the news items being propagated as two types of nodes. A directed edge is drawn from the news item to the source of the item. Other directed edges are drawn from the nodes re-propagating the news items to the news item. A sample re-propagation graph is shown in Fig 1.

The construction of such graph would enable the use of standard community detection and core decomposition algorithms to detect the presence of similar users. If the users indulge in mutual propagation of messages, they would form cycles in the retweet graph of minimum length 4. The presence of a large number of such cycles can be detected automatically. The graph also enables the use of standard PageRank algorithm to rate the quality of the news items and credibility of source nodes. Detection of manipulation of PageRanks would help us to segregate messages of poor quality as well as misinformation.

3.4 Proposed Taxonomy of Semantic Attacks in Online Social Networks

OSNs are susceptible to both spread of disinformation and misinformation. Using detailed analysis of different data sets, we propose a taxonomy for Semantic attacks in

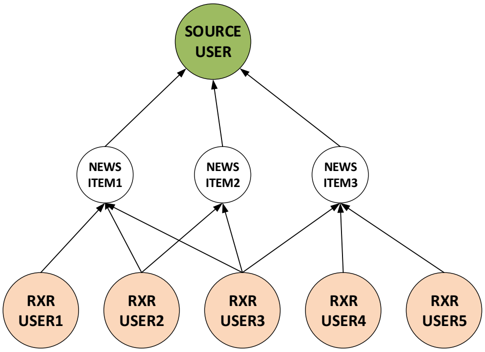


Fig. 1. Sample retweet graph drawn as a bipartite graph with user nodes and message nodes

OSNs based on the intention of the source of messages and the methodologies adopted for the spread of information. The proposed taxonomy is given in Fig 2. We give the details of each in the subsequent sections.

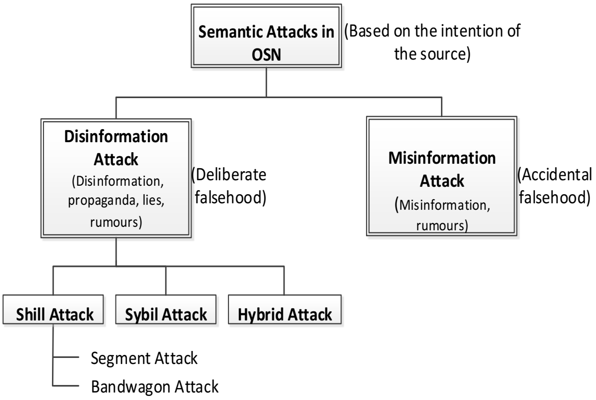


Fig. 2. The taxonomy of Semantic Attacks in Online Social Networks

3.5 Detection of Patterns of Information Propagation

In order to examine the propagation of news items as measured by the retweets in the graph, we use community detection algorithms based on modularity. We used Gephi software [4] for all visualisations of graphs in this paper. We show the distribution of communities for two data sets - Egypt and Syria in Fig 3. As we see in the figure, the propagation of news items is marked by large number of isolated communities with very few communities having connections with others. So identification of communities where the misinformation may be spreading would indicate whether the spread is

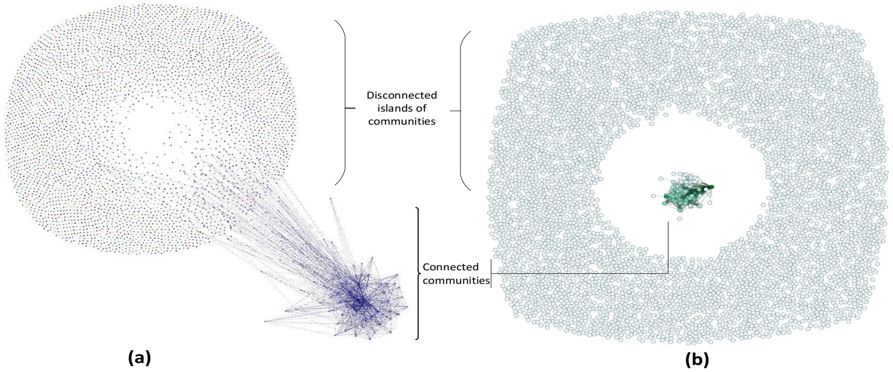


Fig. 3. Distribution of communities in the (a) Egypt and (b) Syria data sets

limited to a single community or it has spread across communities. The number of user nodes in the communities would also indicate the users who are 'infected' with the news items or 'susceptible' to them.

3.6 Detection of Sybil Attacks

In Recommender systems, Sybil attacks involve creation of multiple profiles by the same user to nuke or push up the recommendations of target items. Multiple profiles can be created in OSNs with the aim to propagate a target news item in the network. With URL shorteners available, the messages may seem different, but they would be pointing to the same web page. Detecting these false messages using content analysis of all the tweets would be computationally expensive.

The detection of cores in a graph can be done using k-core decomposition algorithm [2]. We applied k-core decomposition algorithm to the day wise distribution of tweets in the Andhra data set. The Fig 4 shows the inner most cores of Day 4 and Day 5 of the period of collection. The appearance of a set of core users increased the coreness of the Andhra data set from 5 to 9 on these days, indicating the extreme level of mutual communication between the new set of users. A visual analysis would also show that the names of all the users start with 'aum' and the retweets shown in the figure point towards a single URL, which was misinformation[1].

3.7 Detection of Shill Attacks

Shill attacks in Recommender systems are carried out with the aim of increasing the ratings of target items. In the case of OSNs, this could take the form of preferentially propagating certain news items. Different types of semantic attacks noticed in the analysis of data have been categorised based on their similarity with those in Recommender systems.

[1] http://pic.twitter.com/mLXgltMW36

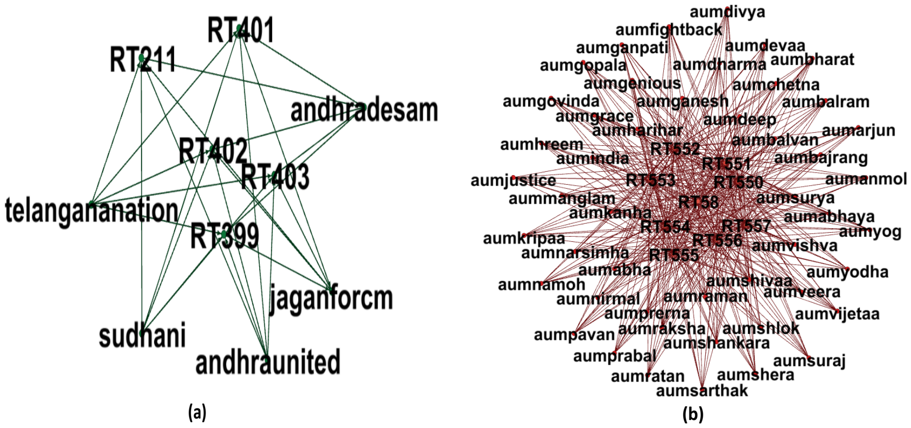


Fig. 4. View of inner most core of the retweet graph of Andhra data set on (a) Day 4 and (b) Day 5 of the period of collection

Bandwagon Attack. In this type of attack, the propagation of news item is carried out by making certain keywords in the message similar to the ones trending at the point in time. In Twitter, *hashtags* are used to tag news pertaining to a single event or entity to form a common thread. The target news item which needs to be propagated in the network is referred in the message using a shortened URL and the rest of the 140 characters permissible in the tweet are used to include all popular hashtags. The aim would be to get bracketed with all the top trending topics and thus make it visible to wider population and ensure its propagation. The promoted news item normally would have no similarity with the other items. The example of sybil attack shown in Fig 4 also is a type of Bandwagon Attack only as seen in the contents of the tweets[2].

Segment Attack. This type of attack is targeted at a specific group or entity. Twitter sees lots of activity during crisis events like earthquakes, acts of terrorism etc. Often the handling of the event by the administration or acts by a community are commented upon by lots of users in social networks. Targeted attacks to change the perception for or against a section or community is often witnessed. Core analysis of the Bodhgaya and MyJihad data sets as shown in Fig 5 are examples of these type of attacks. The message nodes begin with ‘RT’ in their names. The users and the messages in both the data sets were similar in nature. Efforts were noticed to spread false information against a particular community to incite perceptions against them in the wake of bomb blasts at Bodhgaya[3].

3.8 Hybrid Attack

This combines the other types of methodologies to carry out semantic attack. In this, the attackers make use of Bandwagon and Segment attack methodologies coupled with

[2] https://twitter.com/aumsatya/status/381027667350269952/photo/1

[3] http://twitpic.com/aa0wg1

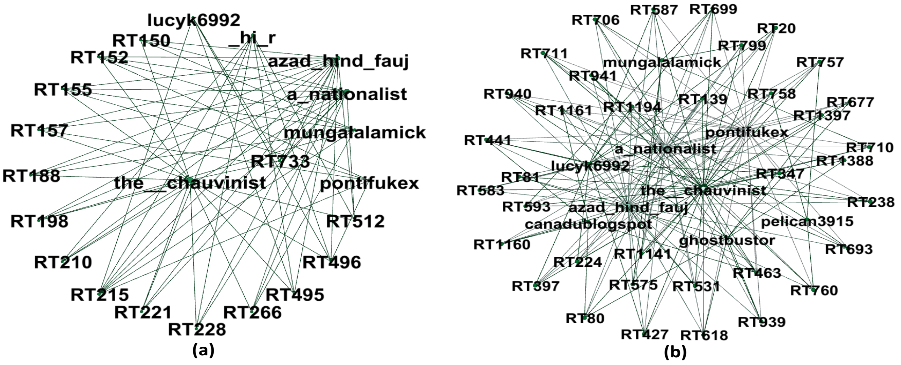


Fig. 5. The tweets in the core of the (a) Bodhgaya and (b) MyJihad data sets are examples of *Segment attacks*

Sybil attacks to augment the authenticity of their messages and ensure more effective spread.

3.9 Misinformation Attack

This attack involves the actual spread of misinformation related to the topic. The source possess good knowledge of the subject and probably believe in them and wants others to be also convinced about his view . This could be in the form of unsubstantiated news, speculation or biased information about the subject. The level of acceptance of such news items would indicate greater reluctance on the part of users to accept the information and more questions being asked about them in the tweets. Examples of this were seen in all the data sets[4].

4 Algorithm for Detection of Spread of Misinformation

A methodology which would enable us to sequentially detect deliberate spread of misinformation is given this section. We applied k-core decomposition algorithm iteratively to the retweet graphs in all the data sets given at Section 3.1, till we have the inner most nodes. We segregate the user nodes and message nodes in the inner most core. We then delete these nodes and edges from the graph and apply the k-core decomposition algorithm again. We iteratively repeat the process till we are left with only nodes which have become isolated after deletion of its edges. These nodes are part of the 1-shell. The aim is to identify the colluding nodes in the propagation of information and separate them. These nodes would form part of the inner cores. Once these nodes are separated, the re-propagation graph should display more even distribution of message nodes and

[4] https://twitter.com/naveentirthani/status/382706087452884993/photo/1

Algorithm 1. Methodology for detection of misinformation and colluding nodes

```
Obtain data for the specific ‘context’ from the social network
Construct re-propagation graph with user nodes and message nodes
while (NumberofRemainingNodes != 0) do
    Estimate the k-value of the innermost core of the re-propagation graph using k-core decomposition algorithm
    Segregate user nodes and message nodes of the k-core in the re-propagation graph
    Delete the nodes and edges in the k-core from the graph
end while
Identify the set of nodes in the inner most cores whose removal would result in a sharp increase in the remaining nodes of the next outer core
The segregated inner most cores would have the user and message nodes involved in collusion for the spread of information
```

user nodes. We give the steps of the proposed methodology in Algorithm 1 and outline them in Fig. 6. The results obtained are shown in Fig 7.

We established the ground truth of our data using human annotation. In Fig 7, the figure on the left is the identification of the inner cores of all data sets. The figure on the right shows the precision of identifying misinforming nodes in the innermost core in the graphs. The source nodes in the inner most cores were identified to spread misinformation with an accuracy around 0.95. The true positives identified from all the inner cores of graphs constituted over 90% of the nodes spreading false information in the complete data set. More importantly, the results prove the effectiveness of the methodology to quickly identify the nodes spreading deliberate false information by the analysis of under 5% of the nodes in the data sets with an accuracy of over 90%. Hence, we believe that the ‘innermost-to-outermost’ methodology of isolating nodes from the retweet graph of the social network data sets is very efficient and suited for development of an effective social media monitoring system.

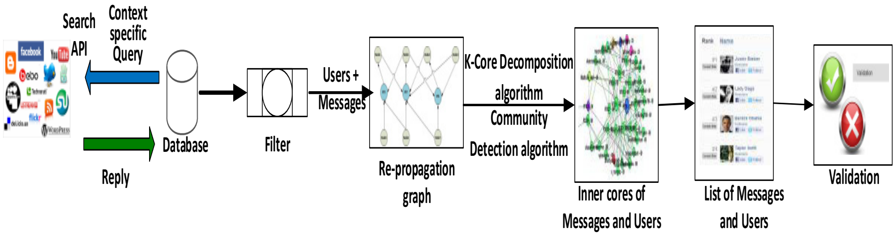

Search API
Context specific Query
Reply
Database
Filter
Users + Messages
Re-propagation graph
K-Core Decomposition algorithm
Community Detection algorithm
Inner cores of Messages and Users
List of Messages and Users
Validation

Fig. 6. Steps for detection of misinformation and colluding nodes

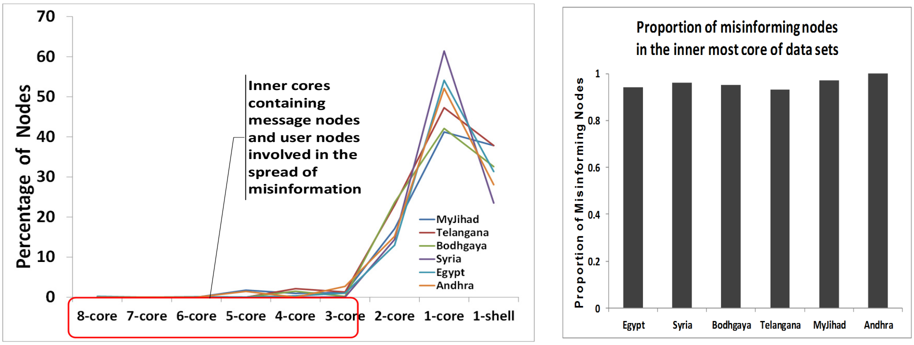


Fig. 7. Core-wise distribution of retweets and user nodes

5 Conclusion and Future Work

The detection of spread of misinformation in OSNs is crucial in tackling the menace of misuse of social media. By reversing the traditional approach, and thus estimating the potential of the tweets to affect the target population using automated means prior to carrying out semi automated semantic analysis, we would be able to effectively detect and prevent the spread of misinformation in an acceptable time frame. We are in the process of extending the algorithm to enable a user to take informed decisions while forwarding information he has received. Automated estimation of credibility of information being propagated in OSNs would enable us to prevent the spread itself.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Alvarez-Hamelin, J.I., Dall'Asta, L., Barrat, A., Vespignani, A.: k-core decomposition: A tool for the visualization of large scale networks. arXiv preprint cs/0504107 (2005)
3. Bakshy, E., Hofman, J.M., Mason, W.A., Watts, D.J.: Everyone's an influencer: quantifying influence on twitter. In: Proceedings of the 4th International Conference on Web Search and Data Mining, pp. 65–74. ACM (2011)
4. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks. In: International AAAI Conference on Weblogs and Social Media, pp. 361–362 (2011)
5. Burke, R., Mobasher, B., Williams, C., Bhaumik, R.: Classification features for attack detection in collaborative recommender systems. In: Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining, pp. 542–547. ACM SIGKDD (2006)
6. Fallis, D.: A conceptual analysis of disinformation. In: iConference. Chapel Hill, NC (2009)
7. Hawksey, M.: Twitter Archiving Google Spreadsheet TAGS v5. JISC CETIS MASHe: The Musing of Martin Hawksey (EdTech Explorer) (2013), http://mashe.hawksey.info/2013/02/twitter-archive-tagsv5/ (last accessed December 29, 2013)

8. Karlova, N.A., Fisher, K.E.: Plz RT: A social diffusion model of misinformation and disinformation for understanding human information behaviour. Information Research 18(1), 1–17 (2013)
9. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: Proceedings of the 19th International Conference on World Wide Web, pp. 591–600. ACM (2010)
10. OMahony, M.P., Hurley, N.J., Silvestre, G.C.: Attacking recommender systems: The cost of promotion. In: Proceedings of the Workshop on Recommender Systems, in Conjunction with the 17th European Conference on Artificial Intelligence, pp. 24–28. Citeseer (2006)
11. Ratkiewicz, J., Conover, M., Meiss, M., Gonçalves, B., Patil, S., Flammini, A., Menczer, F.: Truthy: mapping the spread of astroturf in microblog streams. In: Proceedings of the 20th International Conference Companion on World Wwide Web, pp. 249–252. ACM (2011)
12. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web, pp. 851–860. ACM (2010)
13. Solis, B.: The information divide: The socialisation of news (2010), `http://www.briansolis.com/2010/02/the-information-divide-the-socialization-of-news-and-dissemination/` (last accessed December 29, 2013)
14. Stahl, B.C.: On the difference or equality of information, misinformation, and disinformation: A critical research perspective. Informing Science: International Journal of an Emerging Transdiscipline 9, 83–96 (2006)
15. Starbird, K., Palen, L.: (How) will the revolution be retweeted?: Information diffusion and the 2011 Egyptian uprising. In: Proceedings of the International Conference on Computer Supported Cooperative Work, pp. 7–16. ACM (2012)
16. Zhang, F.: Reverse bandwagon profile inject attack against recommender systems. In: Proceedings of the 2nd International Symposium on Computational Intelligence and Design (ISCID), pp. 15–18. IEEE (2009)
17. Zhang, F.: Analysis of bandwagon and average hybrid attack model against trust-based recommender systems. In: Proceedinga of the 5th International Conference on Management of e-Commerce and e-Government, pp. 269–273. IEEE (2011)

Enhanced Security of PHR System in Cloud Using Prioritized Level Based Encryption

D. Sangeetha, Vaidehi Vijayakumar, Valliammai Thirunavukkarasu,
and Aiswarya Ramesh

Information Technology, Madras Institute of Technology, Chennai, India
dsangeetha@mitindia.edu, vaidehi@annauniv.edu,
{valliammai991,aisu1810}@gmail.com

Abstract. Cloud Computing has emerged as one of the vital part of the IT industry and it requires users to entrust their valuable data to cloud providers and so, there has been increasing security and privacy concerns on outsourced data. However there are more privacy concerns when the data involved is related to health. The current trend is that all the sectors are now moving to paperless management setup reducing the manual work and increasing the efficiency in both technical and management perspective. Similarly, the traditional health records are now being exported to cloud platform for continuous availability and easier management. This opens up the important problem of security when handling the personal data. To mitigate such security risks, proper cryptographic measures must be taken. Proper delegation and revocation mechanisms must be applied in case of sharing the records. There is a need for categorizing the data based on the sensitivity level of the health records, since encrypting all the records using the same mechanism will not be fair and also paves the way for intruders to decrypt all the records if the algorithm is found. To achieve fine-grained and scalable data control for Personal Health records (PHR), we leverage Prioritized Level Based Encryption (PLBE) techniques to encrypt each patient's PHR file, the PHR also includes both text and image data like x-rays and scanned images. Therefore separate encryption techniques have to be enforced for text and image data. We also focus on multiple data owner scenario and divide the users in the PHR system into multiple security domains that reduces key management complexity for both owners and users.

Keywords: Cloud Computing, Data Security, Personal Health Records, Prioritized Level Based Encryption, Sensitivity Analyzer.

1 Introduction

In recent years, health records have been computerized and digitized; in compliance with the modern technology. Personal Health Record (PHR) contains data related to the patients' current health conditions, medical images, medication, diagnosis and other private information like security numbers, family medical history etc.,

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 57–69, 2014.

These PHRs enables the patients and doctors to manage the details easily by making them available online (Ming Li et al. 2013). The PHR owners enjoy the full right of accessing their records anywhere and anytime making storage and retrieval more efficient.

The demands of computing infrastructure have been increasing a lot nowadays, and with the advent of cloud computing, these demands can be met, with optimized cost and increased efficiency. This makes more worthy to shift the PHR services into cloud, which will be more advantageous in the aspects of increased storage capacity, reduced operational and installation costs and increased privacy by sharing the records only within a particular organization. Specialized data center needs are eliminated by moving the data into cloud, and this attracted more vendors to provide PHR services using Cloud technology.

While it will be very useful for having the PHRs online, potential security risks exists. It opens up one's private details to be available in the hands of the intruders. Since the patients placing the PHR will be losing the physical control over to their own personal health data. The data becomes viable for both the insider and outsider attacks due to the high value of the Personal Health Information (PHI). For the former example of insider attack, is the famous incident of a Department of Veterans Affairs database containing sensitive PHI of 26.5 million military veterans, including their social security numbers and health problems was stolen by an employee who took the data home without authorization. For the latter, since cloud computing is an open platform, the servers are subjected to malicious outside attacks. For example, Google has reported attacks to its Gmail accounts in early 2010. Even though there exists regulations such as the Health Insurance Portability and Accountability Act of 1996 (HIPAA) the risks involved in storing them, in a third party server are still indispensable.

To deal with these potential risks, there is a need for security measures like password protection for individual users, encrypting and storing data etc., but these services can only be provided by the PHR service providers and they cannot be initiated by the PHR owners, thus opening up the privacy problems. The patients can generate their own private keys and encrypt their PHRs. In addition to the traditional access control mechanisms, the PHR owners must have the right to share their records selectively to their friends, well-wishers; care-takers etc. They should have the proper delegation and revocation mechanism that makes key generation and management easier.

In this work, a novel solution is proposed for categorizing and encrypting the PHR file which includes text and multimedia data like images in a more trusted way as compared to the previous existing solutions. Multi owner settings are also considered through the separation of domains viz., personal and public domain. In the former, the users could be the patients as described above and in latter, the beneficiaries include doctors, researchers etc. The whole system is categorized into these two domains, and the PHRs are prioritized according to their criticality level and proper encryption algorithm is applied based on the concepts of Prioritized Level Based Encryption (PLBE).

2 Related Works

This work concentrates on the encryption of the data stored in the cloud. Traditional public key encryption algorithms incur high cost and prove difficult in key generation and management techniques. In (Goyal et al. 2005) paper on Attribute Based Encryption (ABE), the data is encrypted under the set of attributes, and proper keys are delegated to the corresponding users.

Secure Sharing of PHR Using ABE

In this paper (Ming Li et al. 2013) the solution is proposed for the secure sharing of PHR attributes. The PHR system is classified as personal and professional Domain. Encryption under multi-owner settings is also taken care. The users of the professional domain are distributively managed by the various cluster heads, they take care of the key generation and key management facilities. Each cluster head manages the disjoint subset of user attributes and role based access control is provided for the users.

Fine Grained Data Access Control

(L. Ibraimi et al. 2007; V. Goyal et al. 2006)proposed a solution for the fine grained data access control in the PHR System by implementing the Cipher text policy Attribute based Encryption (CP-ABE). They assume the single trusted authority (TA) for the whole system. This resulted in the load bottleneck and also suffers from the key escrow problem. But they have overcome the problem of efficient sharing of PHR attributes.

Securely Outsourcing Large Scale System of Linear Equations

(Wang et al. 2013) describe solving the large scale system of linear equations securely in a cloud environment. A random number generator is used to generate the set of random inputs. The inputs are formulated into set of linear equations and these equations are solved iteratively. It uses the homomorphic encryption scheme to securely solve the linear equations. This technique is used as the base for encryption of images in the PHRs.

Revocable ABE

There are as such problems in revoking the permissions as in granting. Proper revocation techniques should be handled otherwise there is a possibility of misuse of data. (Ming Li et al. 2013) have overcome this by on- demand revocation. A proper suite of access control mechanisms have been proposed (Luan Ibraimi et al. 2009). Role based access encryption is been enforced for granting permissions and sharing attributes. This also suffers from the single TA problems and hence the key escrow problems are pre dominant here. For the revocation purposes the user needs to be online.

Classification of Attributes

In (M. Pirretti et al. 2010), a hierarchical classification of attributes of the PHR system has been enforced leading to the combinations of different levels. This ensures minimized time for key generation and key management techniques. But it does not account for the classification of PHR documents based on the criticality levels. So we are trying to enforce the different encryption techniques based on the criticality level of the PHR documents.

3 Security Mechanisms

The security mechanisms used here for providing security are ABE and Paillier Cryptosystem.

ABE

In the existing system (Ming Li et al. 2013), the PHR's of the patients have been classified into personal domain and public domain. Hierarchy of attributes for each record has been defined. Then the record is encrypted under a certain set of attributes as specified. The keys for a record that has been encrypted under personal domain can be delegated to other persons based on the owner's wish. Similarly a user in the public domain can access a certain set of patient records if those keys are delegated earlier to him. The same procedure goes for the revocation of the keys to the granted users. For public domain, Multi-Authority Attribute Based Encryption (MA-ABE) is used and for personal domain Attribute Based Encryption (ABE) algorithm is used. The keys are also delegated to the emergency department for emergency purposes. Access control policy is defined for the patient who wants to delegate their record to their friends.

Paillier CryptoSystem

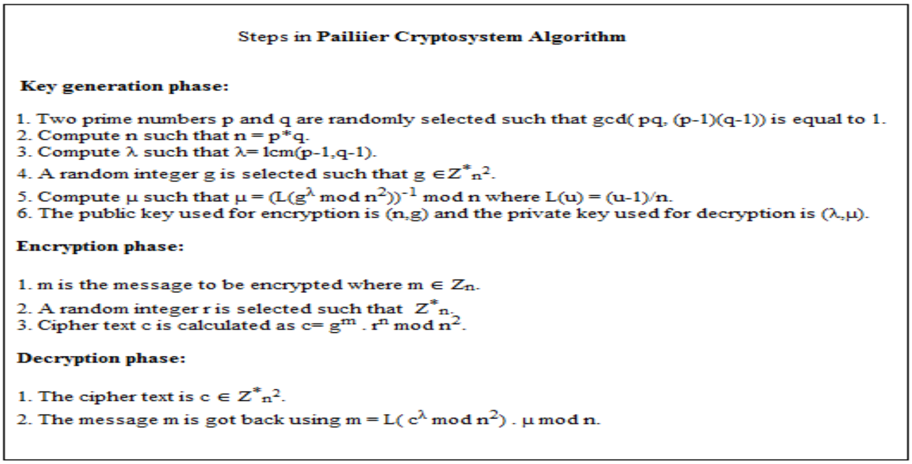
Paillier Cryptosystem, is a asymmetric encryption technique under the homomorphic encryption scheme. The fig. 1 describes the key generation, encryption and decryption stages of the Paillier Cryptosystem.

4 Proposed System

In this section, we describe our novel patient centric data access system based on the sensitivity level in the cloud environment. Here we classify all the PHR records based on the criticality level or sensitivity level and then apply the proper encryption algorithms based on the examined sensitivity level.

Problem Definition

We consider a PHR system with multiple PHR owners and PHR users. The owners refer to the patients who have full control over the data in their PHR file and users refer to the persons who can only read the data from others file. A central server is provided by the PHR service providers where all the records of the PHR owners are stored. Users can access the PHR files through the web. The PHR documents can be classified based on the criticality level and stored in the hierarchical manner.



Steps in Pailiier Cryptosystem Algorithm

Key generation phase:

1. Two prime numbers p and q are randomly selected such that gcd(pq, (p-1)(q-1)) is equal to 1.
2. Compute n such that n = p*q.
3. Compute λ such that λ= lcm(p-1,q-1).
4. A random integer g is selected such that $g \in Z^*{}_{n^2}$.
5. Compute μ such that $\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n$ where L(u) = (u-1)/n.
6. The public key used for encryption is (n,g) and the private key used for decryption is (λ,μ).

Encryption phase:

1. m is the message to be encrypted where $m \in Z_n$.
2. A random integer r is selected such that $Z^*{}_n$.
3. Cipher text c is calculated as $c= g^m \,.\, r^n \bmod n^2$.

Decryption phase:

1. The cipher text is $c \in Z^*{}_{n^2}$.
2. The message m is got back using $m = L(c^\lambda \bmod n^2) \,.\, \mu \bmod n$.

Fig. 1. Paillier Cryptosystem Steps

Security Model

We consider a honest but curious cloud server. That means the server will try to find out as much secret information in the stored PHR files as possible, but they will honestly follow the protocol in general. The server may also collude with a few malicious users in the system. On the other hand, some users will also try to access the files beyond their privileges. For example, a pharmacy may want to obtain the prescriptions of patients for marketing and boosting its profits. To do so, they may even collude with other users.

Overview of the Proposed Modules

To enforce the above mentioned constraints of security, we divide the whole system into public and personal domain as mentioned before. The security measures of our PLBE are imposed over them.

In the personal domain the PHR owners can encrypt, decrypt and share their record with their friends. In following the encryption methodology it would not be fair to encrypt all patients' record using the similar algorithm. For example a person suffering from a normal flu and a person suffering from cancer cannot be encrypted under the roofs of the same cryptographic algorithm. The PHRs must be classified based on their criticality or sensitivity level and different encryption algorithms must be enforced at different level of the Sensitivity Analyzer (SA) methodology. The proposed solution from collection to revocation of PHRs is as follows

Data Collection and Storage

The PHR data is collected from various hospitals with patients having problems of different levels of criticality. For Data Storage we use the database MongoDB. MongoDB is a horizontally scalable open-source document oriented database implemented in C++ and developed by 10gen. MongoDB bridges the gap between key/value stores and traditional RDBMS (Relational Database Management Systems).

MongoDB is schema free and stores data in a binary form of JSON called BSON (Binary JSON) and supports replication. MongoDB is a NoSQL database, which stores the data in terms of the document. It is proved that NoSQL Databases offer a high performance and availability compared to relational databases. Finally, the MongoDB database is integrated with the cloud infrastructure with a connector.

Cloud Environment

Accessing the PHRs via cloud based systems is more effective. Generally, cloud provides efficient access of data. This influences the application of cloud in PHR system. This application provides flexible as well as reliable support to the hospital management services. It is easy to provide uploading and downloading of data in/from the database which is stored and processed in cloud environment. So, cloud environment is able to fill all the prerequisites that are to be needed for PHR system. So we have setup the Hadoop environment for the same enhancing scalability and reliability.

Data Classification

Data classification is done prior to the data storage. During this phase the data is to be categorized on the basis of CIA (Confidentiality, Integrity, and Availability). The client who wants to send the data for storage needs to give the value of C (confidentiality), I (integrity), A (Availability). The value of C is based on level of secrecy at each junction of data processing and prevents unauthorized disclosure. The value of I based on how much assurance of accuracy is provided, reliability of information and unauthorized modification is required. The value of A is based on how frequently it is accessible.

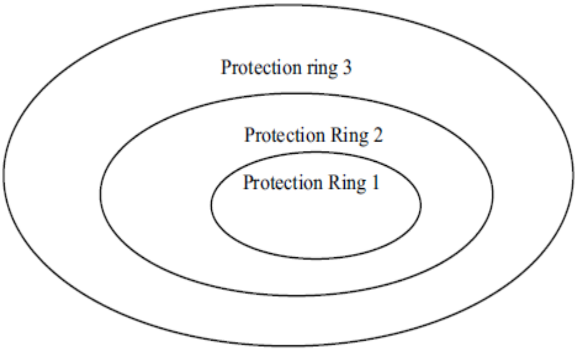


Fig. 2. The proposed framework for the Data Classification in different protection rings

The PHR documents are analyzed by the SA and it is classified into three protection rings as described in fig 2, where, Protection Ring 3 corresponds to PHR files with low criticality level. Protection Ring 2 corresponds to PHR files with moderate criticality level. Protection Ring 1 corresponds to PHR files with extreme criticality level. Thus encrypting data in the Protection Ring 3 has to be taken much more importance than Protection Ring 2, than the data in Protection Ring 1. For this purpose SA imposed with PLBE is encouraged.

Attribute Identification

The traditional encryption algorithm involves, encrypting the whole PHR document. In PLBE, a particular attribute(s) can be encrypted and decrypted. For this purpose the PHR document is classified into various attributes. The attributes can be classified into a hierarchical tree structure as mentioned in the fig 3. The shaded boxes correspond to the personal domain and other boxes represent the public Domain.

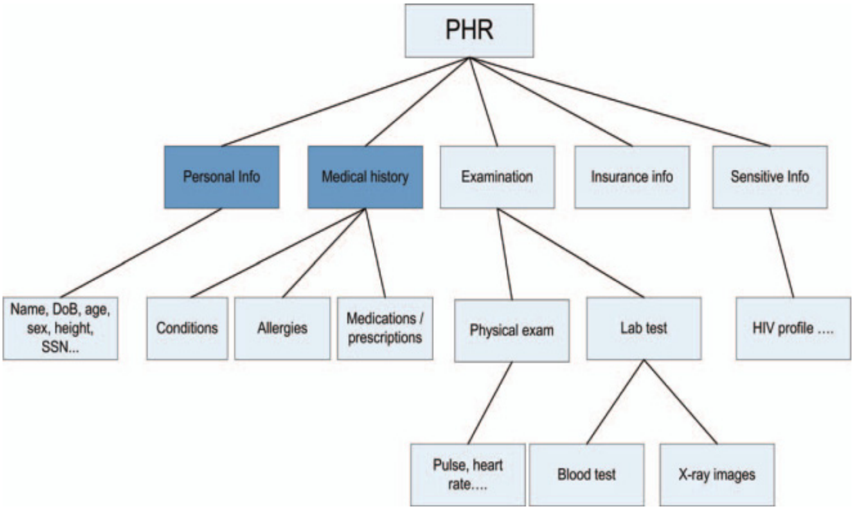
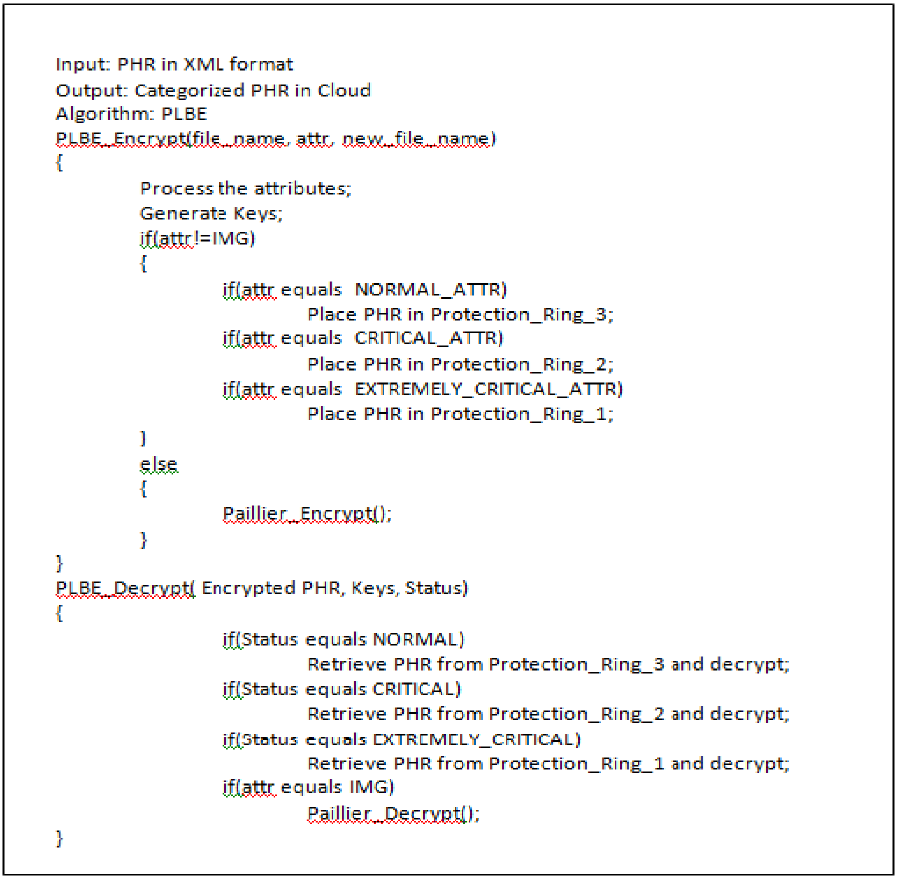


Fig. 3. Level based Classification of PHR Attributes

Encryption and Decryption

The PHR document is encrypted based on a particular attribute, which can be predefined by the PHR owner or can be mentioned at the time of encryption. For example, in the document having the attributes as in fig 3, it is time consuming to encrypt the whole file, while there is a need for encrypting only few critical attributes like personal info, Medications/Prescriptions, HIV Profile etc. So the main idea is to encrypt only the critical attributes, thus saving time and effort involved. The system also provides the option of encrypting the text and images separately. For encrypting the images, first the image has to be converted into pixels, in the next stage the pixel array is converted into matrix based on the image dimensions and the key generation, encryption and decryption steps are followed as in fig. 1. Fig. 4 gives the pseudo code of the whole framework.



```
Input: PHR in XML format
Output: Categorized PHR in Cloud
Algorithm: PLBE
PLBE_Encrypt(file_name, attr, new_file_name)
{
          Process the attributes;
          Generate Keys;
          if(attr!=IMG)
          {
                    if(attr equals  NORMAL_ATTR)
                              Place PHR in Protection_Ring_3;
                    if(attr equals  CRITICAL_ATTR)
                              Place PHR in Protection_Ring_2;
                    if(attr equals  EXTREMELY_CRITICAL_ATTR)
                              Place PHR in Protection_Ring_1;
          }
          else
          {
                    Paillier_Encrypt();
          }
}
PLBE_Decrypt( Encrypted PHR, Keys, Status)
{
                    if(Status equals NORMAL)
                              Retrieve PHR from Protection_Ring_3 and decrypt;
                    if(Status equals CRITICAL)
                              Retrieve PHR from Protection_Ring_2 and decrypt;
                    if(Status equals EXTREMELY_CRITICAL)
                              Retrieve PHR from Protection_Ring_1 and decrypt;
                    if(attr equals IMG)
                              Paillier_Decrypt();
}
```

Fig. 4. Pseudo code of PLBE Algorithm

Break Glass Access

In emergency situations, the possibilities are very less for the patient to reveal the keys. To overcome this difficulty, the set of temporary keys are delegated to the Emergency Department (ED). In such situations of emergency, the emergency staff from the ED can authenticate themselves with the cloud service provider and can get access to the PHR files. Once the emergency situation is under control, the PHR owner can revoke those temporary keys and can assign a new one from that emergency staff.

User Revocation

The system is flexible, by providing the user, the revocation rights. The PHR owners have the options of sharing their records with friends, care-takers etc. If they feel to revoke those permissions from those whom they have shared already, they can do by just updating the policies another time revoking the granted permissions.

5 PLBE Framework in PHR System

In this section, we will look into the inner details of the PLBE framework and how it has been applied in the PHR system.

Access Policy

In the basic usage, we consider a special class of access policy conjunctive normal form (CNF), $P := (A1 = a1, 1) \vee \cdots \vee (A1 = a1, d1) \wedge \cdots \wedge (Am = am, 1) \vee \cdots \vee (Am = am, dm)$ *m is the total* number of attribute types and *A* refers to the attributes. The Access policy of a PHR can be described as in the example below.

P1 :="(profession=physician) ∧ *(specialty = internal medicine)* ∧ *(organization= hospital A)"*

The access policies are defined and only the user with the corresponding keys can only be able to decrypt the specific PHR.

Architecture of PLBE Framework

The PHR owners encrypt their files and upload them in the cloud, in which the file gets stored in the MongoDB database. Before the encryption, the file is classified according to its sensitivity level by the SA module. When there is a need the PHR users can download the file and decrypt it with the right keys. The whole framework is described in fig. 5.

Sensitivity Analyzer

The SA classifies the PHRs based on the sensitivity level and places them on their corresponding protection rings. Based on this, different types of encryption algorithms are carried out. The data in, Protection Ring 3 is encrypted using the normal file encryption algorithm. PHRs in Protection Ring 2 are encrypted under PLBE algorithm and data in Protection Ring 1 is double encrypted using PLBE thus ensuring enhanced security.

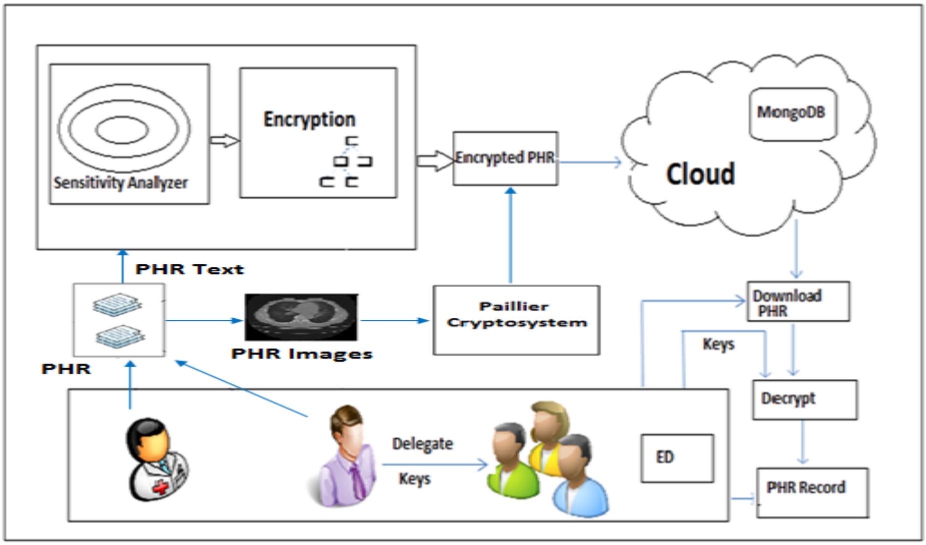
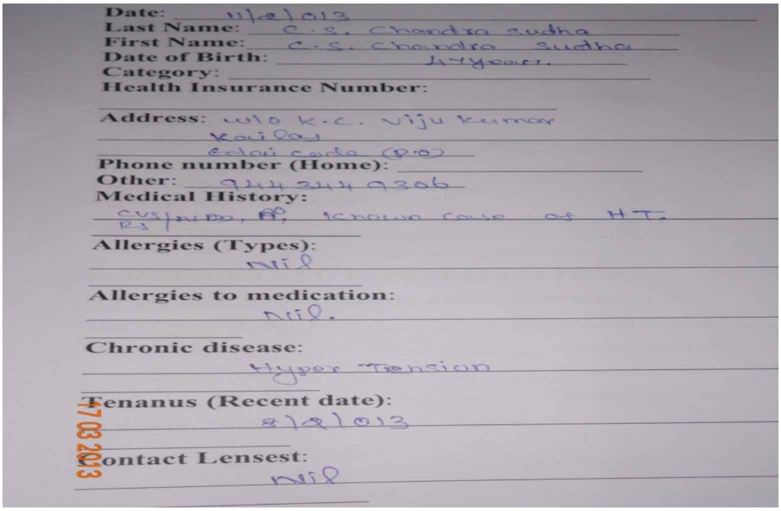


Fig. 5. Architecture of PLBE Framework

6 Implementation

We have implemented this work on the Hadoop environment, thus ensuring scalability and reliability. Hadoop is a framework for running applications on a large cluster built of commodity hardware. The hadoop framework transparently provides both the reliability as well as data motion. Hence the Hadoop Distributed File System (HDFS) has been appropriately chosen. For the experimentation purpose, a model of private cloud setup is taken and results are obtained. The programming language chosen is java. A sample PHR record is shown in fig. 6. The screenshot of encrypted files in the hadoop environment is shown in the fig. 7. The corresponding encrypted file, for the one shown in fig. 6 is ‘encrypted.xml’ in fig.7. The PHR files have been classified under the sensitivity analyzer module and stored on the different protection rings using the appropriate algorithms.



Date: 11/2/013
Last Name: C.S. Chandra sudha
First Name: C.S. Chandra Sudha
Date of Birth: 47 years.
Category:
Health Insurance Number:

Address: w/o K.C. Viju Kumar
Kailas
Edai code (P.O)
Phone number (Home):
Other: 944 344 9306
Medical History:
CVS/RS/ABD, [illegible], known case of H.T.

Allergies (Types):
Nil

Allergies to medication:
Nil.

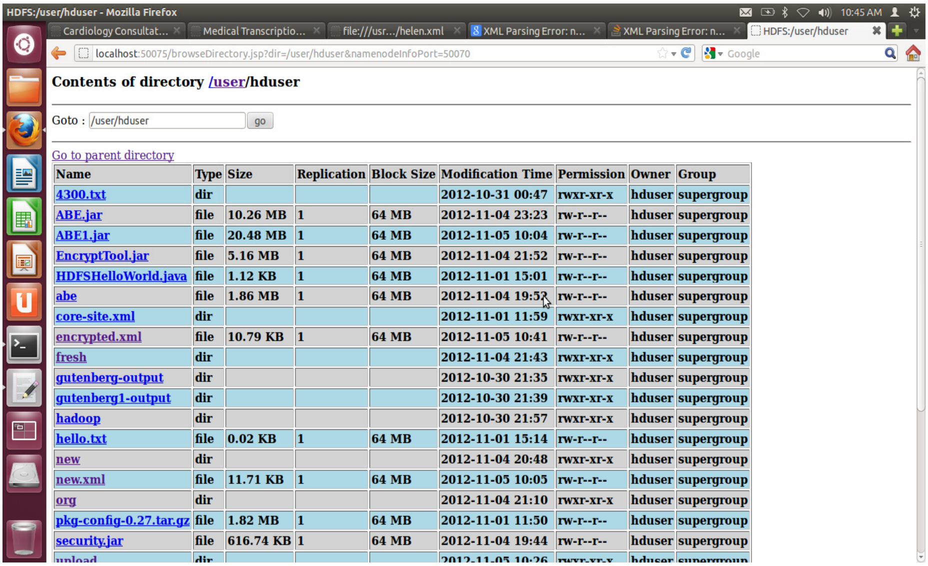
Chronic disease:
Hyper Tension

Tenanus (Recent date):
8/2/013

Contact Lensest:
Nil

Fig. 6. Sample PHR File

A sample image and its encrypted pixel results are shown in the fig. 8 and fig. 9 respectively. Since any PHR might contain sensitive information like patient scan images and other information as in fig. 8 they should also be encrypted to assure privacy. Fig 9 shows the implementation output of the encrypted image stored in the form of encrypted pixels.



Contents of directory /user/hduser

Goto : /user/hduser go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
4300.txt	dir				2012-10-31 00:47	rwxr-xr-x	hduser	supergroup
ABE.jar	file	10.26 MB	1	64 MB	2012-11-04 23:23	rw-r--r--	hduser	supergroup
ABE1.jar	file	20.48 MB	1	64 MB	2012-11-05 10:04	rw-r--r--	hduser	supergroup
EncryptTool.jar	file	5.16 MB	1	64 MB	2012-11-04 21:52	rw-r--r--	hduser	supergroup
HDFSHelloWorld.java	file	1.12 KB	1	64 MB	2012-11-01 15:01	rw-r--r--	hduser	supergroup
abe	file	1.86 MB	1	64 MB	2012-11-04 19:52	rw-r--r--	hduser	supergroup
core-site.xml	dir				2012-11-01 11:59	rwxr-xr-x	hduser	supergroup
encrypted.xml	file	10.79 KB	1	64 MB	2012-11-05 10:41	rw-r--r--	hduser	supergroup
fresh	dir				2012-11-04 21:43	rwxr-xr-x	hduser	supergroup
gutenberg-output	dir				2012-10-30 21:35	rwxr-xr-x	hduser	supergroup
gutenberg1-output	dir				2012-10-30 21:39	rwxr-xr-x	hduser	supergroup
hadoop	dir				2012-10-30 21:57	rwxr-xr-x	hduser	supergroup
hello.txt	file	0.02 KB	1	64 MB	2012-11-01 15:14	rw-r--r--	hduser	supergroup
new	dir				2012-11-04 20:48	rwxr-xr-x	hduser	supergroup
new.xml	file	11.71 KB	1	64 MB	2012-11-05 10:05	rw-r--r--	hduser	supergroup
org	dir				2012-11-04 21:10	rwxr-xr-x	hduser	supergroup
pkg-config-0.27.tar.gz	file	1.82 MB	1	64 MB	2012-11-01 11:50	rw-r--r--	hduser	supergroup
security.jar	file	616.74 KB	1	64 MB	2012-11-04 19:44	rw-r--r--	hduser	supergroup
upload	dir				2012-11-05 10:26	rwxr-xr-x	hduser	supergroup

Fig. 7. Encrypted files in Hadoop environment

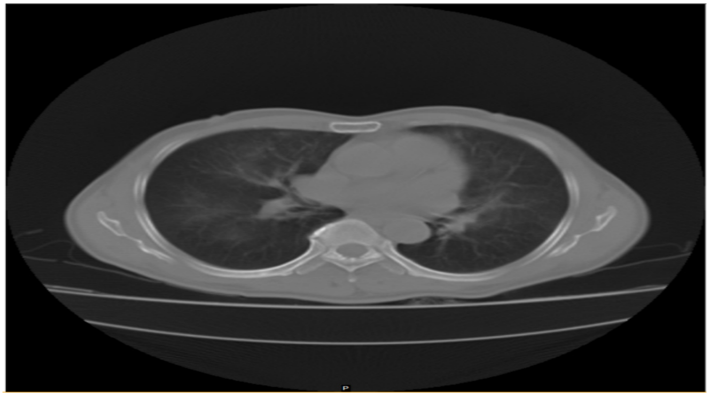
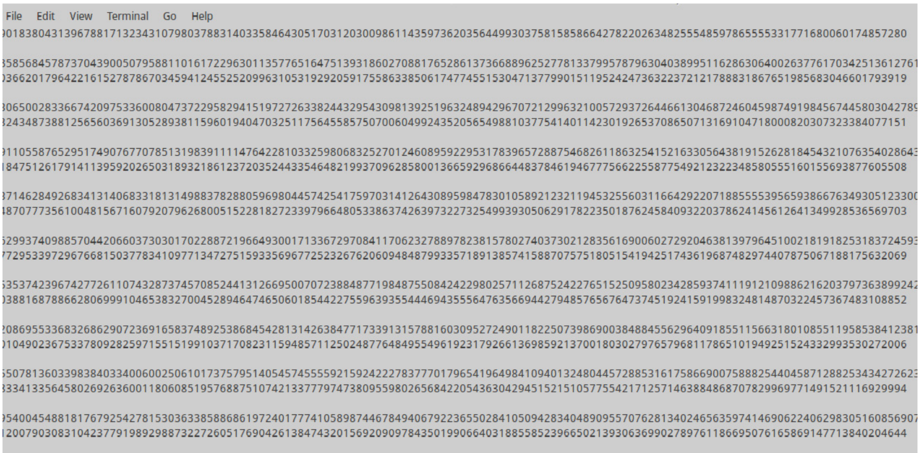


Fig. 8. Sample image to be encrypted



```
90183804313967881713234310798037883140335846430517031203009861143597362035644993037581585866427822026348255548597865555331771680060174857280

3585684578737043900507958811016172296301135776516475139318602708817652861373668896252778133799578796304038995116286306400263776170342513612761
036620179642216152787867034594124552520996310531929205917558633850617477455153047137799015111952424736322372121788831867651985683046601793919

3065002833667420975336008047372295829415197272633824432954309813925196324894296707212996321005729372644661304687246045987491984567445803042789
32434873881256560369130528938115960194047032511756455857507006049924352056549881037754140114230192653708650713169104718000820307323384077151

9110558765295174907677078513198391111476422810332598068325270124608959229531783965728875468261186325415216330564381915262818454321076354028643
18475126179141139592026503189321861237203524433546482199370962858001366592968664483784619467775662255877549212322348580555160155693877605508

3714628492683413140683318131498837828805969804457425417597031412643089598478301058921232119453255603116642922071885555395659386676349305123300
48707773561004815671607920796268005152281827233979664805338637426397322732549939305062917822350187624584093220378624145612641349928536569703

5299374098857044206603730301702288721966493001713367297084117062327889782381578027403730212835616900602729204638139796451002181918253183724593
77295339729676681503778341097713472751593356967725232676206094848799335718913857415887075751805154194251743619687482974407875067188175632069

5353742396742772611074328737457085244131266950070723884877198487550842422980257112687524227651525095802342859374111912109886216203797363899242
03881687886628069991046538327004528946474650601854422755963935544469435556476356694427948576567647374519241591998324814870322457367483108852

2086955336832686290723691658374892538684542813142638477173391315788160309527249011822507398690038488455629640918551156631801085511958538412381
01049023675337809282597155151991037170823115948571125024877648495549619231792661369859213700180302797657968117865101949251524332993530272006

5507813603398384033400600250610173757951405457455559215924222783777017965419649841094013248044572885316175866900758882544045871288253434272623
33341335645802692636001180608519576887510742133777974738095598026568422054363042945152151057755421712571463884868707829969771491521116929994

9540045488181767925427815303633858868619724017774105898744678494067922365502841050942834048909557076281340246563597414690622406298305160856907
12007903083104237791989298873227260517690426138474320156920909784350199066403188558523966502139306369902789761186695076165869147713840204644
```

Fig. 9. Encrypted Pixel Results

7 Results and Discussions

We have compared the time taken by the normal file encryption algorithm and our PLBE. The results show that, PLBE consumes less time compared to traditional algorithms, thus proving to be time efficient as in fig.10. It also increases the service response time, and service availability. Key generation and key management complexities are reduced. Since the system is placed over cloud environment, scalability increases, thus reducing the storage, setup, operation and communication costs

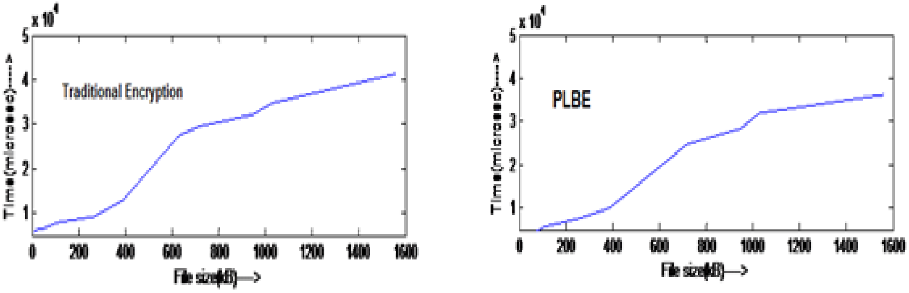


Fig. 10. Comparison of traditional encryption algorithm and PLBE

8 Conclusion

In this paper, we have proposed a novel solution for sharing of the PHR in a secure way. We utilize PLBE to encrypt the PHR data not only by users, but also various users from public domains with different professional roles, qualifications and affiliations and also on-demand user revocation is provided. This framework addresses the unique challenges brought by multiple PHR owners and users, where we reduce the complexity of key management while enhancing the privacy guarantees.

References

Li, M., Yu, S., Ren, Y.Z.K., Lou, W.: Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption. IEEE Transactions on Parallel and Distributed Systems 24 (2013)

Wang, C., Ren, K., Wang, J., Wang, Q.: Harnessing the cloud for securely outsourcing large-scale systems of linear equations. IEEE Transactions on Parallel and Distributed Systems 24(6), 1172–1181 (2013)

Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute based encryption. In: Shands, D. (ed.) Proceedings of the 28th IEEE Symposium on Security and Privacy, pp. 321–334 (2007)

Ibraimi, L., Tang, Q., Hartel, P., Jonker, W.: Efficient and provable secure ciphertext-policy attribute-based encryption schemes. In: Bao, F., Li, H., Wang, G. (eds.) ISPEC 2009. LNCS, vol. 5451, pp. 1–12. Springer, Heidelberg (2009)

Prasad, P., Ojha, B., Shahi, R.R., Lal, R., Vaish, A., Goel, U.: 3 dimensional security in cloud computing. In: 3rd International Conference on Computer Research and Development (ICCRD), vol. 3, pp. 198–201 (2011)

Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine grained access control of encrypted data. In: CCS 2006, pp. 89–98 (2006)

Ibraimi, L., Petkovic, M., Nikova, S., Hartel, P., Jonker, W.: Ciphertext-policy attribute-based threshold decryption with flexible delegation and revocation of user attributes (2009)

Li, M., Yu, S., Cao, N., Lou, W.: Authorized Private Keyword Search over encrypted data in Cloud Computing. In: 31st International Conference on Distributed Computing Systems (2011)

Ibraimi, L., Asim, M., Petkovic, M.: Secure Management of Personal Health Records by applying attribute based encryption. In: 6th International Workshop on Wearable Micro and Nano Technologies for Personalized Health (2009)

Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: Secure attribute based systems. Journal of Computer Security 18(5), 799–837 (2010)

Wan, Z., Liu, J., Deng, R.H.: A hierarchical attribute based solution for flexible and scalable access control in cloud computing. IEEE Transactions on Information Forensics and Security 7(2) (2012)

online, At risk of exposure – in the push for electronic medical records, concern is growing about how well privacy can be safeguarded (2006)

MM-SOAR: Modified SOAR Protocol Using Multiple Metrics for Wireless Mesh Network

Anuradha Jadiya and Deepak C. Karia

Abstract. Wireless Mesh Networks (WMNs) are bringing revolutionary changes in the field of wireless communication. Routing protocols and the cost metric play an important role in the performance of WMN. Currently most common metric used in Opportunistic Routing is Expected Transmission Count (ETX) based on the link's error rate and have drawback of lower throughput and higher overhead. In this paper we propose a new opportunistic routing protocol, Multiple Metric-Simple Opportunistic Adaptive Routing (MM-SOAR) with improved throughput. We introduce a new routing metric, Multiple Metric Cost (MMC), integrating three metrics: 1) Available bandwidth, 2) Residual energy and 3) Expected Transmission Count (ETX). MM-SOAR is simulated on OMNET++ and our extensive evaluation shows that MM-SOAR gives better performance than traditional Destination Source Routing (DSR) protocol and SOAR under a wide range of scenery.

1 Introduction

Wireless Mesh Network (WMN) has been gaining considerable attention from industry and academia in recent years. WMNs are self-organized multi-hop networks comprising of stationary and mobile nodes. Nodes in a WMN consist of mesh routers and mesh clients [1]. There exist three main types of architectures of WMNs: 1) Hybrid WMNs, 2) Client WMNs and 3) Infrastructure/Backbone WMNs.

Routing Metric is a crucial issue for the design of WMN for achieving good performance and reliability. Researchers have proposed a number of performance metrics for WMNs. It has also been shown that integrating multiple performance metrics into a routing protocol is effective for attaining optimal performance. Here in our work we introduce a Multiple Metric Cost (MMC) a new routing metric integrating link's available bandwidth, node's residual energy and ETX to optimize the performance of SOAR.

To demonstrate effectiveness of MM-SOAR, we have implemented this protocol in the OMNET++ network simulator. Using extensive evaluation, we show that proposed method performs better than DSR as well as SOAR under a wide range of scenery.

The rest of the paper is organized as follows: Section 2 reviews the related work pertaining to this research work. Section 3 describes the proposed Multiple Metric SOAR (MM-SOAR) protocol. Simulation work and results are discussed in section 4. Finally, we conclude in Section 5.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 70–78, 2014.

2 Related Works

Initially routing protocols for wireless networks were based on best path routing, i.e., they select one fixed route before the transmission starts. These are known as proactive routing protocols for example Destination-Sequenced Distance-Vector Routing (DSDV), Cluster Head Gateway Switch Routing (CGSR) and Optimized Link State Routing Protocol (OLSR). Another class of routing is known as reactive routing or on-demand methods. The route discovery process is initiated when the source needs a route to destination. Example of reactive routing protocols is Dynamic Source Routing (DSR) protocol and Ad-hoc On Demand Distance Vector (AODV) protocol.

Both proactive and reactive protocols treat wireless links as point-to-point wired links and ignore the unique broadcast nature of the wireless medium so these schemes are not best suited for WMN, and Opportunistic Routing (OR) protocols came into existence. OR conceptually follows three steps [2]:

- Broadcast a data packet to forwarding nodes.
- Select the best forwarding node using coordination protocol.
- Forward the data packet.

Consider an example as shown in Fig. 2 [2], where the source S is sending a packet to the destination D through nodes R1, R2, R3 and R4 and the number of each edge indicates the sequence of events. First, S broadcasts a packet. R2, R3 and R4 successfully received it but R1 failed. Then, R2, R3 and R4 run a coordination protocol and decide that R2 forwards the packet to D.

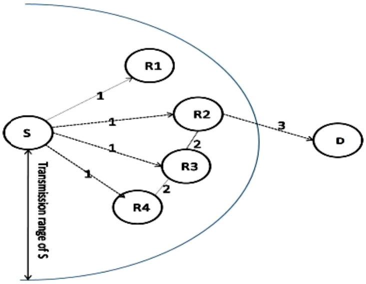


Fig. 1. Working of OR Protocol [2]

2.1 Opportunistic Routing Design Issues

Optimized route in OR is achieved through:

- **Forwarding node selection or candidate selection:** All nodes in the network must run an algorithm for selecting and sorting the set of neighboring nodes (forwarders) that can better help in the forwarding process to a given destination.
- **OR metric**: In order to accurately select and prioritize the forwarding nodes, OR algorithms require a routing metric. Example ETX is used by many OR protocols.

- **Candidate coordination:** It is the mechanism used by the forwarding nodes to assign priorities based on routing metric used. Node with minimum cost with destination given highest priority. Coordination protocols are classified as timer based, token based and network coding based [2].

2.2 SOAR Protocol

SOAR [3] is a straight forward representation of opportunistic routing, i.e. the selection of the next forwarding node is done after the actual data transmission.

Consider a packet appearing as one of a set of mesh nodes running the SOAR protocol. If the destination of the packet is not the node itself, the default path is calculated by Dijkstra's shortest path (DSR) algorithm. Then, a list of forwarding nodes ordered by priority is calculated and added to the packet. The highest priority node is the one with the lowest remaining path cost (i.e. With lowest ETX) to the final destination [4].

Receivers drop packets if they are not in forwarding list. The highest priority node immediately forwards the packet, while the others start forwarding timers. If a node overhears the forwarding of a packet by a higher priority node for which it has a forwarding timer running, it cancels its timer and drops its copy of the packet. Or else, as soon as the forwarding timer elapses, it forwards the packet by itself [4].

To support opportunistic routing, each node maintains a routing table of the following format: (destination, default path, forward List), where the default path is the shortest path from the current node to the destination and the forwarding list includes a list of next-hop nodes that are eligible to forward the transmission [3]. Here the main interest area is how forwarding nodes are selected using ETX metric. Forwarding list selection algorithm consists of two steps. First, a sender selects an initial forwarding list based on the default path. Then it further limits the number of forwarding nodes to minimize duplicate transmissions.

Consider the calculation of the forwarding list from the current node i to the destination d. Let ETX(a,b) be the ETX of the link a-b, and let pETX(a,b) be the total ETX of the shortest path from a to b. When node i is on the default path, i select the forwarding nodes using algorithm: I [3], [4] given below:

```
Algorithm: I
    Forwarders = ( )
for each node x in topology
do
    if pETX(x, d) < pETX(i, d) and ETX(i, x) < threshold
            add x to forwarders
    prune (forwarders)
    done
```

With that, a node is added to the forwarding list, if and only if two conditions match. First, the ETX to the destination from the candidate node has to be less than from the current node. Second, the ETX of the link from the candidate to the current node must be lower than a given threshold. At last, the forwarding list is pruned, meaning it is ensured that the link ETX of all node pairs are also within the before mentioned threshold.

Now, algorithm: II [3], [4] is used, whenever i is not on the shortest path.

```
Algorithm: II
1:    Forwarders = ()
2:    closest = none
3:    for each node x in shortest path
4:    do
5:    if pETX(i, x) < pETX (closest,x)
6:              closest = x
7:    done
8:    if pETX (closest, d) < pETX(i, d)
9:              add closest to forwarders
10:   for each node x in forwarding list of closest
11:   do
12:   if pETX(x, d) < pETX(i, d) and ETX(i,x) < threshold
13:             add x to forwarders
14:   done
```

Here, lines 3 to 7 seek for the node on the shortest path that is metrically closest to i. If closest is at the same time close to the ultimate destination, then it gets added to the list of forwarders (lines 8 to 10). Finally, in lines 11 to 14, the forwarding list of the closest is obtained using the previously examined algorithm [4]. From that list, all nodes that are both closer to the destination and within the threshold to cur are added to the new list. SOAR outperforms much better than DSR, but its performance can be improved further if multiple metrics are used to calculate the default path and forwarding nodes.

3 MM-SOAR

Most commonly used metric in OR is ETX for example Extremely Opportunistic Routing (ExOR), MORE and SOAR all uses ETX. But the performance of ETX becomes low because it neither distinguishes links with different bandwidths nor considers data packet sizes. Second thing is the single metric fails to realize or describe the QoS due to several factors which also includes mobility, power loss etc.

Hence merely establishing a path using ETX is not justified. Therefore we propose a new metric Multiple Metric Cost (MMC) which consider three routing metrics: 1) link's available bandwidth, 2) residual energy of nodes and 3) ETX. Multiple-Metric cost is calculated as:

$$MMC = ETX \text{ x } (1/BWmac) \text{ x } (1/E) \tag{1}$$

Where, BW_{mac} = Link's available bandwidth and E= Node residual energy.

ETX have drawback of high overhead. We are integrating residual energy, in MMC to reduce overhearing nodes at the physical layer and this will reduce overhead.

3.1 Available Bandwidth (BW_{mac}) Calculation

The Bandwidth can be conceptually defined as the number of packets, which a link can accommodate. In this work we have used MAC bandwidth. A MAC bandwidth is the available throughput sensed by a Node in a link. Mathematically it is defined as:

$$BW_{mac} = \alpha \text{ x } (1/T_d) \tag{2}$$

Where, α is a weighted factor between $0 \leq \alpha \leq 1$ and T_d is transmission delay. To calculate transmission delay T_d, a synchronized system is assumed. Hello message is broadcasted periodically to all nodes. When the hello message arrives at a neighbor node, the difference between the time when a packet leaves the sender node and the time the packet arrives at the destination node is measured. This measured time is $T_{d.}$ We may also consider average Round Trip Time (RTT) as transmission delay.

3.2 Energy Calculation

We use the residual energy model of the physical layer, to calculate E at instant t [5].

$$E(t) = E(t -1) - (E_{rx} P_{rx} \text{len}(P_{rx})d^n) - (E_{rx}P_{tx}\text{len}(P_{tx})d^n) \tag{3}$$

Where, E_{rx} energy lost per reception and E_{tx} amount of energy needed for every transmission in bit/m. P_{rx} and P_{tx} are the number of packets received and transmitted by the node respectively between interval t and t-1. E(t-1) is the energy of the node at (t-1) instance. d is the distance over which packets are traversed and is determined as Euclidian distance. n is a constant between 1 and 2 and is considered 2 here.

After calculating MMC, following steps are used in MM-SOAR for forwarding packets:

- Find out default path using a DSR algorithm with minimum MMC.
- Find out initial forwarding node list based on default path using the algorithm: I as discussed in section 2, but use MMC in place of ETX.
- Then use algorithm: II for pruning of forwarding nodes to limit duplicate transmission, with MMC in place of ETX.
- Now store default path and forwarding nodes in the routing table of every node for forwarding the packets to the destination.

4 Simulation and Results

4.1 Simulation Setup

MM-SOAR using proposed metric is simulated in OMNET++. We use an ETX-based shortest-path routing protocol (also known as DSR) and SOAR with ETX as a baseline comparison. For simulation essential parameters are listed in table 1.

Table 1. Simulation Parameter

Simulation Parameters	Parameter Value
Propagation Model	Raleigh fading
Physical layer standard	802.11
Radio Sensitivity	-90dBm
Simulation Time	900s
Radio Range	250m
MAC Protocol	Link layer. IEEE80211. MAC
Network Size	950m x 950m
Mobility Model	Random Waypoint
Packet Size	512 Bytes
No of nodes	35 (To measure impact of link's transmission rate)
Topology	Random Topology

4.2 Simulation Results

4.2.1 Single Flow

We used random topology to evaluate performance under single flow. We varied transmission rate from 100 pkts/s to 2000 pkts/s to measure the performance of the protocol (i.e. Throughput, control overhead, and latency). The total bandwidth of the network is 11MHz and Nodes are communicating within 4 hops. So effective bandwidth at each node is around 2.5Mbps. Each packet is 512 Bytes and hence bottleneck transmission for each node is 2.5 Mb/512B = 2000 packet approx. Fig. 2 compares throughput of MM-SOAR with SOAR and shortest path (DSR). It shows that MM-SOAR gives better throughput than SOAR and DSR.

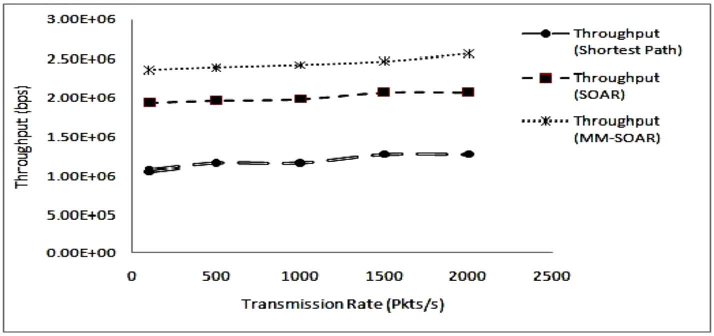


Fig. 2. Throughput comparison of MM-SOAR, SOAR and shortest path routing with variable transmission rate

Comparison of control overhead is shown in Fig. 3, it shows that shortest path has highest overhead, and MM-SOAR has the lowest overhead among three. Both MM-SOAR and SOAR uses broadcast transmissions and has no ACK overhead, while shortest path routing uses unicast transmissions and incurs ACK overhead. MM-SOAR includes residual energy in cost metric, and this limits the number of overhearing nodes in physical layer and a lower overhead is achieved.

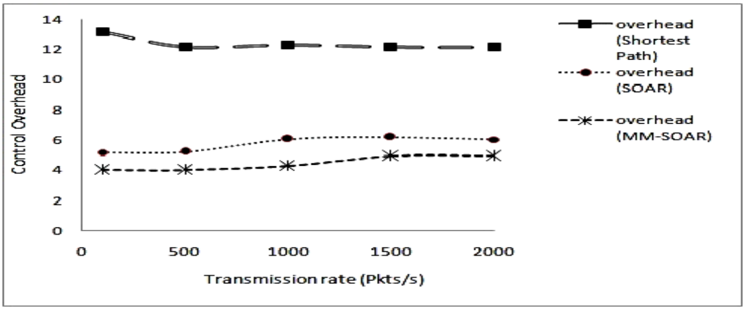


Fig. 3. Overhead comparison of MODIFIED SOAR, SOAR and shortest path routing with variable transmission rate

Latency or delay is a very important parameter in a communication network. Fig. 4 shows a latency comparison of all three protocols.

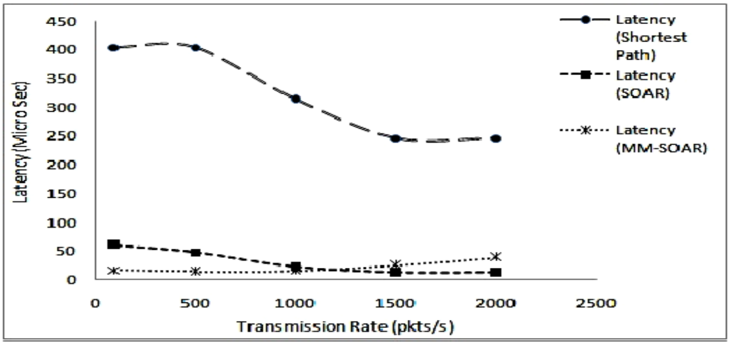


Fig. 4. Latency comparison of MM-SOAR, SOAR and shortest path routing with variable transmission rate

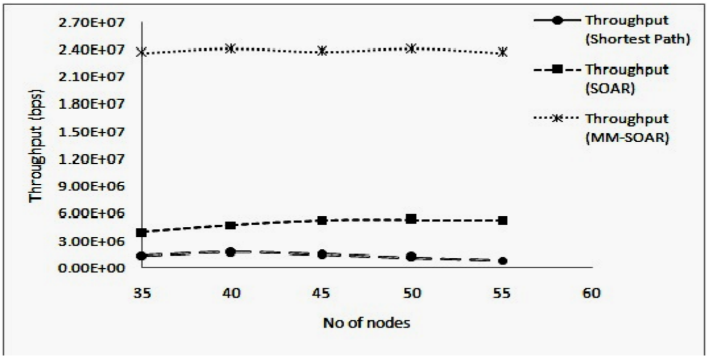


Fig. 5. Throughput comparison of MM-SOAR, SOAR and shortest path routing with variable node density

We observe that latency of MM-SOAR is less than shortest path but slightly more than SOAR protocol for lower transmission rate, because three metrics are used for selecting default paths and forwarding nodes. Our future objective is to reduce this delay at least equal to SOAR or even lesser.

Node density also affects the performance of routing protocol. We vary number no nodes from 35 to 55 and observed the performance. Fig. 5 shows the effect on throughput of all three protocols.

Fig. 6 shows control overhead measured by varying the no of nodes. Here also modified SOAR proves that it has better performance over SOAR and shortest path.

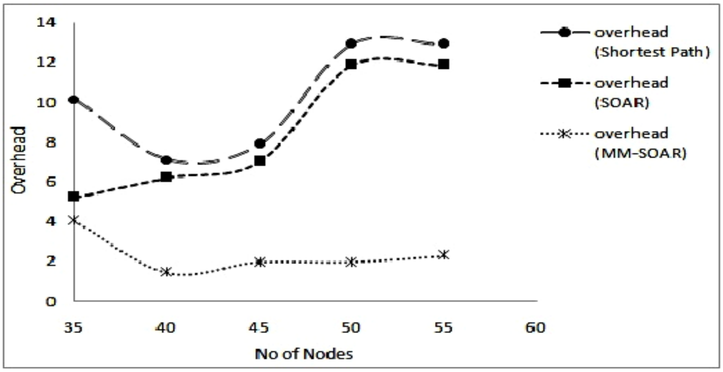


Fig. 6. Overhead comparison of MM-SOAR, SOAR and shortest path routing with variable node density

4.2.2 Multiple Flows

Next we evaluate the performance of multiple flows by randomly choosing sources and destination pairs in a network of 35. We measured packet delivery ratio (PDR) of MM-SOAR with SOAR and DSR. Fig. 7 proves that MM-SOAR performs better than SOAR and DSR for multiple flows also.

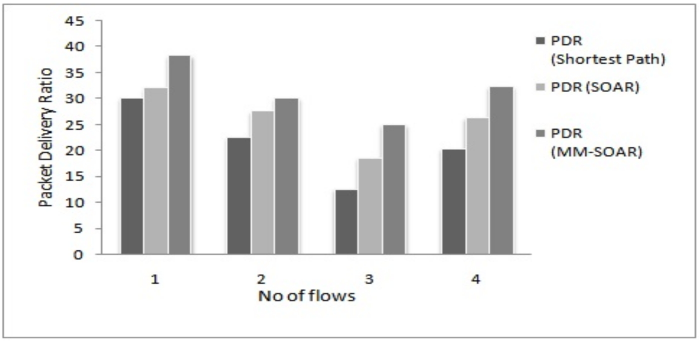


Fig. 7. Overhead comparison of MM-SOAR, SOAR and shortest path routing with variable node density

5 Conclusion

In this work, we have presented Multiple-Metric SOAR protocol under the class of opportunistic routing for Wireless Mesh Networks. We proposed a new metric MMC integrating three link cost metric i.e. ETX, available bandwidth and energy to calculate default path and forwarding nodes. We used the OMNET++ simulator to show that MM-SOAR have better throughput and lesser overhead when compared with ETX based SOAR and DSR. One drawback with MM-SOAR is that it has a higher latency than SOAR protocol. The latency of MM-SOAR could further reduce with enhanced default path and forwarding node selection, which we plan to investigate as part of our future work.

References

[1] Akyildiz, W.I.F., Wang, X.: Wireless mesh networks: a survey. Computer Networks Journal 47, 445–487 (2005)

[2] Hsu, C.J., Liu, H.I., Seah, W.K.G.: Survey paper: Opportunistic routing - a review and the challenges ahead. Computer Network 55(15), 3592–3603 (2011)

[3] Rozner, E., Seshadri, J., Mehta, Y., Qiu, L.: SOAR: Simple Opportunistic Adaptive Routing Protocol for Wireless Mesh Networks. IEEE TMC (2009)

[4] Sarrar, N.: Implementation and evaluation of an opportunistic mesh routing protocol. Master's thesis, Technische Universität Berlin (2009)

[5] Sumathi, N., Sumathi, C.P.: Energy and Bandwidth Constrained QoS Enabled Routing for MANETs. In: World Congress on Computer Science and Information Engineering (IEEE CSIE 2011) (2011)

[6] Parissidis, G., Karaliopoulos, M., Baumann, R., Spyropoulos, T.: Routing Metric for Wireless Mesh Network. International Journal (2013)

[7] Draves, R., Padhye, J., Zill, B.: Comparisons of routing metrics for static multi-hop wireless networks. In: Proc.of ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM), August 2004, pp. 133–144 (2004)

[8] Varga, A.: The omnet++ discrete event simulation system. In: European Simulation Multiconference (June 2001)

[9] Zhongliang Zhao, T.B.: OMNeT++ based opportunistic routing protocols simulation: A framework. In: 10th Scandinavian Workshop on Wireless Ad-hoc Networks, ADHOC 2011 (2011)

Clustering Based Routing Strategies for Energy Management in Ad-Hoc Networks

Deepak C. Karia and Shravanee Shinde

Sardar Patel Institute of Technology, Mumbai University, Mumbai, India
{deepakckaria,shravanee14}@gmail.com

Abstract. Clustering is an energy efficient routing technique which is used for ad-hoc network optimization in terms of energy and lifetime. A clustering technique takes into consideration only two parameters that of node-ID and node-Degree. We propose two clustering based routing protocols namely 'Multiple Criterion - Clustering Based Routing Protocol' (MC-CBRP) and 'Residual Energy-Clustering Based Routing Protocol' (RE-CBRP). The proposed protocols take into consideration the node energy along with the other network parameters. These protocols provide an extension to the existing Cluster Based Routing Protocol (CBRP). In RE-CBRP we consider the node energy to be the criterion for the cluster head selection. The node having the maximum residual energy is elected as the cluster head. Whereas MC-CBRP takes into account node ID, node Degree along with node energy to select a cluster head. The results provide comparative performance analysis of RE-CBRP, MC-CBRP, existing CBRP and Dynamic Source Routing (DSR) protocols. Results demonstrate that the network can be efficiently optimized with respect to energy using the RE-CBRP and MC-CBRP as compared to CBRP and DSR.

Keywords: Clustering, ad-hoc networks, routing protocols, CBRP, DSR, RE-CBRP, MC-CBRP.

1 Introduction

The absence of any fixed infrastructure results in an environment where each node may act as a source or destination moving and operating in a distributed peer to peer mode, generating independent data and acting as a router forwarding packets to the next hop allowing multi hop communication to reach the final destination [1].

The development of wireless link has accomplished a high data rate in a high mobility environment, making the wireless devices spend more and more power. Over a period of time the energy of the nodes goes on decreasing and they may die out of the network. With every node dropping out of the network there might be a number of nodes which can no longer be linked to each other resulting in a network partition. The task of finding and maintaining routes in a wireless network is nontrivial since node mobility causes frequent unpredictable topological changes. In a highly mobile situation, the flooding scheme is the most reliable for sending data packets. However,

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 79–90, 2014.

since the link channel and battery power resources are very scarce, more efficient schemes are devised. These schemes require up to date information about the location of nodes. Storage is not a critical issue since memory continues to get less expensive each year. The savings in communication bandwidth and energy come from reporting only to nodes that need particular information. To reduce the transmission overhead for the update of the routing tables after topological changes, it was proposed to divide all nodes into clusters [2].

In general the goal of clustering is efficient utilization of radio channel resources and reduction of overhead. There are traditionally two families of distributed clustering algorithms. One is the lowest-ID algorithm and other is the highest connectivity (degree) algorithm [2], [3]. MC-CBRP proposed also takes into consideration the third criterion of mobile node energy. Hence the selection or re-selection of a cluster head will take place based on an optimum choice between the three parameters namely the node ID, node Degree and node Energy, whereas RE-CBRP takes into consideration only the energy of the mobile nodes. The objective of both the above mentioned protocols are to improve the network lifetime, packet delivery ratio, end to end delay, throughput, normalized routing load and number of nodes dying. The optimum selection of cluster head based on the above mentioned three criteria is done with the help of an equation1 and equation 2 (shown in section 3) relating the said criteria.

2 Related Works

2.1 Dynamic Source Routing Protocol (DSR)

First, The Dynamic Source Routing protocol (DSR) [4], [5] is a simple, efficient and reactive routing protocol designed specifically for use in multi-hop wireless networks. DSR allows the network to be completely self-organizing and self-configuring, without the need for any existing network infrastructure or administration. DSR provides guaranteed loop- free routing, operation in networks containing unidirectional links and very rapid recovery when routes in the network change [6]. There are no periodic hello routing messages which are used by a mobile node to inform neighbors about its presence in the network. In addition, DSR only monitors the operations of the routes in use. Once there is a link failure in a route, the sender of the route is notified immediately [7]. As a result, DSR can quickly adapt to topological changes caused by node movement, which is often observed in a wireless network.

The basic operation of DSR Protocol is divided into two mechanisms:

1. Route Discovery
2. Route Maintenance

1. Route Discovery

When a host in the network wants to send a packet and route to the destination is not available in route cache, host initiates route discovery. The route discovery is done with the help of flooding. The host broadcasts the route request packet to its neighbor.

The route request packet consists of address of destination host as well as the route record which records the host which passes this request. On receiving this request packet, every host checks if it is the destination or route to the destination is known. In both cases, the route from sender to destination is found. This route is replying back to the sender host through the route reply packet [4], [6].

2. Route Maintenance

The disadvantage of the DSR protocol is that the route maintenance mechanism does not locally repair a broken link. The stale route cache information could also result in inconsistencies during the route reconstruction phase. The connection setup delay is higher than in a table-driven protocols. Even though the protocol performs well in static and low-mobility environments, the performance degrades rapidly with increasing mobility. Also, considerable routing overhead is involved due to the source-routing mechanism employed in DSR. Another disadvantage of this protocol is that in case of excessive flooding network clogging may occur.

2.2 Clustering Based Routing Protocol (CBRP)

The Clustering Based Routing Protocol (CBRP) [8], [9] is an on-demand routing protocol in which mobile nodes in the wireless network are divided into clusters. CBRP uses a simple cluster formation strategy. The diameter of the cluster is only two hops and the cluster can be disjoint or overlapping each other. Each cluster elects one of its nodes as clusterhead which is unique for each cluster. This clusterhead maintains all the information about its cluster members and is also responsible for the routing mechanism [8]. Due to the formation of clusters, flooding traffic during route discovery gets minimized and also the process of route discovery gets speed up. The clusterhead knows the addresses of the members in its cluster and each cluster member has a bi-directional link to it. Cluster heads also communicates with each other through the cluster gateway.

The operation of CBRP is divided into following mechanism:

a. Cluster Formation
b. Routing Process
c. Cluster Maintenance

a. Cluster Formation

At any particular instant, mobile nodes in the network are in one of five states: Clusterhead (CH), Cluster Member (CM), Undetermined (U), Cluster Gateway (CGW) and Cluster Guest (CG) [8]. Each node broadcast hello message to its neighbors periodically. When a new node comes up in a network, it enters into undetermined state and broadcast a Hello message. When a clustered gets this message, it responds with a triggered hello message immediately. When the undetermined node gets this message, it sets itself as a cluster member. However, if this undetermined node gets timed out, then it makes itself the clusterhead unless it has a bi-directional link to some of its neighbor otherwise, it remains in undetermined state and repeats the procedure again [9].

b. Routing Process

In CBRP, routing is done using source routing. When a mobile node wants to send data to the destination, it floods the route request packet but only to its neighbor cluster heads. On receiving this packet clusterhead checks whether the destination is in its cluster. If yes, it sends the route directly to the destination node else it sends this request to its adjacent cluster heads. The cluster head's addresses are recorded in the packet so clusterhead discard packet that it has already seen [9]. When the destination node receives this request packet, it replies with the route through the route which is already recorded in the request packet. If the mobile node does not receive a reply from destination within a particular time period, it backs off exponentially before trying to send route request again. While forwarding the packet, if the mobile node detects a broken link to the destination then it notifies the sender node through the route error message and uses a local repair mechanism. In local repair mechanism, when a mobile node finds the next hop is unreachable, it checks to see if the next hop can be reached through any of its neighbors or if hop after next hop can be reached through any other neighbor. If either of the case works, the data packet is sent over the repaired path.

c. Cluster Maintenance

Cluster heads are elected by the individual clusters. It is necessary that the cluster heads must not be changed frequently, hence a non clusterhead node is not allowed to challenge the status of clusterhead even though it satisfy the condition of being an clusterhead[9]. Clusterhead status is changed only when two cluster heads move next to each other, in this case one of the clusterhead has to give up its status of being clusterhead which is dependent on certain rules [8], [3]. CBRP has some limitations and problems which are disadvantages as compared to other protocols. If the network and cluster become too big, the overhead per packet increases due to source routing.

3 Proposed Protocols

After the investigation of the existing clustering algorithms, we found that the selection of the "critical nodes" play a vital role in such hierarchical routing procedures. Faulty selection or selection on the basis of individual factor can lead to inefficient usage of the node energy and ultimately result in network partition and hence shorter network lifetime. Realizing the importance of cluster head (CH) and cluster gateway (CGW), we introduce the proposed clustering scheme with novel cluster head selection procedures in this section. As noted earlier, we have extended the CBRP [8], [9] discussed in the previous sections, to propose a protocol which uses a new and rather efficient cluster head selection process. The motive of introducing a new cluster head selection scheme is to emphasize the importance of nodes acting as cluster head and cluster gateway in a hierarchical routing procedure, such as the clustering based routing. We propose and compare two routing schemes RE-CBRP and MC-CBRP, one with residual energy as the only factor of cluster head selection and another, which considers multiple criteria for selecting the cluster head.

In the first proposed protocol (RE-CBRP), the nodal residual energy is used to determine the cluster head. Certain changes are required to be done in the CBRP to incorporate the residual energy of nodes and thereby in the selection of the head using this residual energy. This scheme uses a two step approach to determine the cluster head:

1. To make each node aware of the energy level of its neighbor nodes.

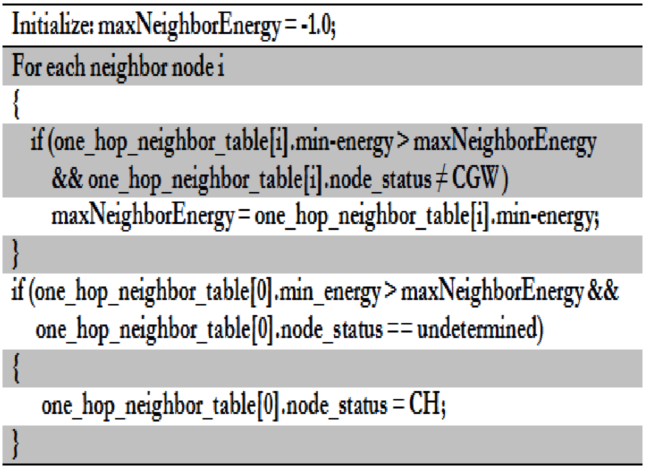
2. To use the energy level information to determine the cluster head.

Step 1: We introduce a term *min-energy* in the one-hop-neighbor table of each node. The field *min-energy* is represented as a fraction of the residual energy of the node to its initial energy. Such a representation introduces computational equanimity in a heterogeneous system. The field *min-energy* is also propagated in the broadcasted HELLO message to make each neighbor node aware of the residual energy level of the other nodes in the cluster. The rest of the packet structure remains same as CBRP.

Step 2**:** The broadcasted information is used by a node, which can either be a node with *undetermined* status or with *clusterhead* status, to determine the cluster head or to give up its cluster head status. The node with the highest value of *min-energy* is selected as the head of the respective cluster. This mechanism also considers the impact of *clustergateway* nodes, which are crucial to prevent network partition in a hierarchical clustering scheme. While the node with maximum *min-energy* is selected as the cluster head, if the node has more than one cluster head's serving at that instance, i.e. if the node is a cluster gateway, then it is not selected as the *clustered* to prevent the expenditure of more energy in the route discovery process.

With RE-CBRP, initially when all nodes are in *an undetermined state*, a node with maximum value of *min-energy* is selected as the *cluster head*. However, at a later instance when a node with *undetermined* status without any cluster head in its range, performs a similar cluster head selection process, it takes care that a node with *clustergateway* status is not selected as the cluster head, as shown in table 1. The intention of such a mechanism is to prevent the overuse of nodes situated in strategically important locations and thereby serving as a cluster gateway. In figure 1, maxNeighborEnergy is the variable used to determine the maximum energy of any neighbor node, one_hop_neighbor_table is the table maintained by each node to identify its immediate neighbor nodes 1 hop away, min_energy and node_status fields in the one_hop_neighbor_table hold the current energy level and status of the nodes. Also, the first element in this table i.e. one_hop_neighbor_table [0] represents the node itself to which the table belongs.

Table 1. Decision Making Algorithm for Node with Undetermined Status



```
Initialize: maxNeighborEnergy = -1.0;
For each neighbor node i
{
  if (one_hop_neighbor_table[i].min-energy > maxNeighborEnergy
     && one_hop_neighbor_table[i].node_status ≠ CGW)
     maxNeighborEnergy = one_hop_neighbor_table[i].min-energy;
}
if (one_hop_neighbor_table[0].min_energy > maxNeighborEnergy &&
   one_hop_neighbor_table[0].node_status == undetermined)
{
    one_hop_neighbor_table[0].node_status = CH;
}
```

Another important phenomenon is when a node with *clusterhead* status gives up its role. A cluster head periodically checks its *neighbor table* to determine whether there are other nodes with *clusterhead* status is within the cluster. If so, then the node determines the cluster head with maximum *min-energy* and compares with its own energy level. If the energy level of the node is less than it gives up its *clusterhead* status and becomes a *clustermember* or *clustergateway* if head_number is equal or greater than 1, respectively as shown in table 2. In table 2, head_number is the number nodes with the *clusterhead* status present within the cluster at a given instant of time.

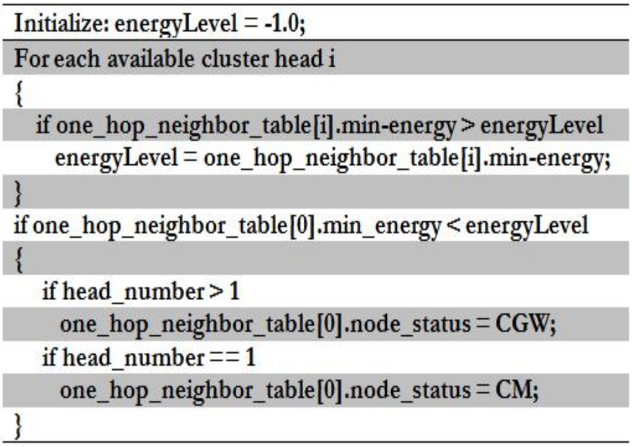
The second proposed protocol (MC-CBRP) aims to enhance the energy efficient clustering scheme in the first version, by considering not only the residual nodal energy but also the node degree and node ID as factors to determine the cluster head. To unify these three different parameters into one factor, which can be ultimately used for the cluster head selection we use an equation to define a factor termed as the head selection factor (HSF). In MC-CBRP, highest preference is given to the *min-energy* followed by node degree and then node ID. Hence, the second version is aimed at combining the advantages of various clustering schemes which individually consider nodal energy, node degree and node ID. This version involves following basic steps:

1. Determining the nodal residual energy of each node in the cluster.

2. Determining the Head Selection Factor (HSF) for each entry in the 1 hop neighbor table.

3. Selection of the Cluster Head on the basis of the HSF.

Table 2. Decision Making Algorithm for Cluster Head



```
Initialize: energyLevel = -1.0;
For each available cluster head i
{
   if one_hop_neighbor_table[i].min-energy > energyLevel
     energyLevel = one_hop_neighbor_table[i].min-energy;
}
if one_hop_neighbor_table[0].min_energy < energyLevel
{
   if head_number > 1
     one_hop_neighbor_table[0].node_status = CGW;
   if head_number == 1
     one_hop_neighbor_table[0].node_status = CM;
}
```

1. *Determining the nodal residual energy of each node in the cluster*

This procedure involves sharing of the information available with each node with every other node in the cluster. This is achieved by broadcasting the HELLO message every HELLO_INTERVAL second. The HELLO message contains the 1-hop and 2-hop neighbor table of the node broadcasting it. As in the first version, the term *min-energy* is included in the 1-hop neighbor table to record and thereby broadcast it to the neighbors. Every receiving node records each entry or updates it, if the entry is already available. Thus, each node in the cluster is aware of the recent energy level of every other node in the cluster and hence, this information can be used to dynamically select the select the cluster head.

2. *Determining the Head Selection Factor (HSF) for each entry in the 1 hop neighbor table*

MC-CBRP uses an equation to relate and arrive at the head selection factor using nodal residual energy, node degree and node ID, with highest preference for residual energy followed by node degree and node ID. In an instant when a node receives a HELLO message with new or updated information, it records this information in the respective neighbor table. While recording any entry in the 1-hop neighbor table, the node also reckons the HSF for the corresponding entry. The HSF is obtained by using equation (1).

$$HSF = e^{(nodeEnergy)} + w_1 * e^{\left(\frac{nodeDegree}{100}\right)} - w_2 * e^{\left(\frac{nodeID}{100}\right)} \qquad (1)$$

The HSF has higher priority for nodes with higher residual energy, higher node degree and lower node IDs. The weighted variables w_1 and w_2 are best chosen as 0.1 and 0.01 respectively.

3. Selection of the Cluster Head on the basis of the HSF

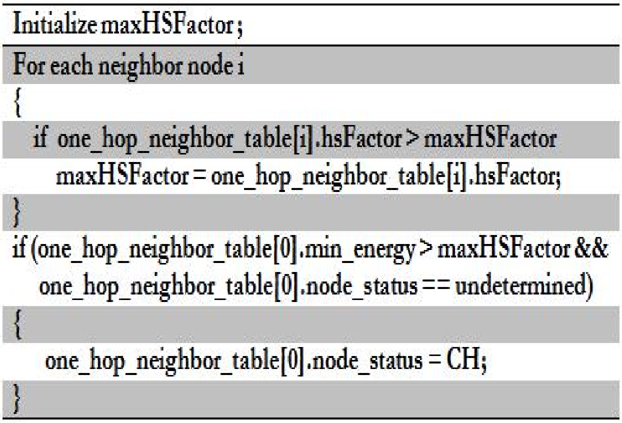
A cluster head is selected in a similar manner as in the first version. A node which is already a *clusterhead* status or has *undetermined* status initiates the cluster head reselection or selection procedure. A node with *undetermined* status examines its 1-hop neighbor table to determine whether it has a cluster head within reach. If yes, then it associates itself to the cluster head. If not, then it checks the HSF of each node to determine the highest value as shown in table 3. If the node itself has the maximum value then it declares itself as the cluster head and changes its status to *clusterhead* and sends out a triggered hello message with the changed status.

If a node with *clusterhead* status upon examining its neighbor table, finds another cluster head within its cluster, then computes the highIDFactor for each cluster head, using the equation (2).

$$HighIDFactor = e^{(nodeEnergy)} + w_1 * e^{\left(\frac{nodeDegree}{100}\right)} + w_2 * e^{\left(\frac{nodeID}{100}\right)} \tag{2}$$

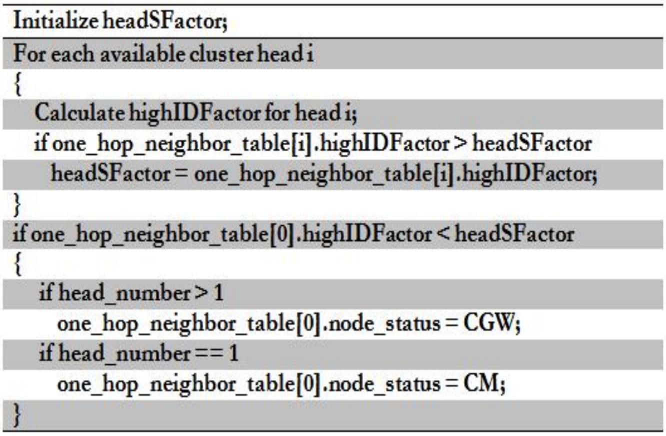
HighIDFactor has higher priority for Cluster heads with higher energy, higher node degree and higher node ID. The weighted variables w_1 and w_2 are best chosen as 0.1 and 0.01 respectively. If the HighIDFactor of the node is less than that of the other head then it releases its *clusterhead* status and changes over to *clustermember* or *clustergateway* depending on the head_number (i.e. Number of cluster heads within the coverage of the node), as shown in table 4. This clustering scheme uses the route discovery mechanism and route maintenance mechanism similar to CBRP, with an assumption that all links are bidirectional. Routing of data packets is done using source routing technique which is used in DSR and CBRP, as discussed in the earlier sections. Although, this scheme introduces some routing overheads compared to the CBRP protocol but it also provides energy efficient routing.

Table 3. Decision making Algorithm for Node with Undetermined Status



```
Initialize maxHSFactor;
For each neighbor node i
{
  if one_hop_neighbor_table[i].hsFactor > maxHSFactor
     maxHSFactor = one_hop_neighbor_table[i].hsFactor;
}
if (one_hop_neighbor_table[0].min_energy > maxHSFactor &&
   one_hop_neighbor_table[0].node_status == undetermined)
{
    one_hop_neighbor_table[0].node_status = CH;
}
```

Table 4. Decision Making Algorithm for Cluster Head



```
Initialize headSFactor;
For each available cluster head i
{
  Calculate highIDFactor for head i;
  if one_hop_neighbor_table[i].highIDFactor > headSFactor
    headSFactor = one_hop_neighbor_table[i].highIDFactor;
}
if one_hop_neighbor_table[0].highIDFactor < headSFactor
{
  if head_number > 1
    one_hop_neighbor_table[0].node_status = CGW;
  if head_number == 1
    one_hop_neighbor_table[0].node_status = CM;
}
```

4 Simulation Environment

The proposed strategies were tested at various node speeds (from 10m/Sec to 40m/Sec) and varied number of node connections i.e. network load independently and the resulting performance indicators are shown in figure [1], [2] and [3]. Table 5 shows the simulation metrics used.

Table 5. Simulation Parameters

Parameters	Values
Simulation Area	1000m x 1000m
Simulation Time	150s
Propagation Model	Two Ray Ground
Mac Layer	IEEE 802.11
Range	150m
Number of Nodes	50
Packet size	512 Bytes
Packet type	UDP, CBR
Rate of Packet Generation	4 packets/Sec
Initial Energy per node	50 J
Receiving Power	1.0 Watt
Transmitting Power	1.2 Watt
Idle Power	0.1 Watt
Hello Message Interval	2.0Sec
Neighbor Table Scan Interval	0.1Sec
Entry Valid Period	4.0Sec

Figure 1 summarizes the lifetime performance of DSR, CBRP, RE-CBRP and MC-CBRP for various node speeds. DSR provides the lowest network lifetime whereas CBRP has improved lifetime as compared to DSR. However, RE-CBRP and MC-CBRP demonstrate elongated network lifetime at higher node speeds in comparison to CBRP. It can be seen that as the node speed increases both the proposed schemes ensure a better network lifetime. This is primarily due to the consideration of residual nodal energy for selection of nodes involved in critical network functions. Further, MC-CBRP facilitates higher network lifetime due to the involvement of node degree and node ID in making routing decisions.

Figure 2 plots the measured average end-to-end delay at various node speeds. It shows that with the increased node speed use of DSR can result in greater delay in delivering data to the destination whereas using CBRP the data can be delivered faster, in general. This is because DSR has the tendency to use stale routes cached in the memory of the nodes. Both RE-CBRP and MC-CBRP present better end-to-end delay performance. In general, the end-to-end delay performance demonstrated by RE-CBRP is better at lower node speed and that of MC-CBRP is better at higher node speed (especially after 20m/Sec). The reason being that MC-CBRP considers the node degree and node ID in addition to other parameters thereby ensuring quick route repair in case of link breakage. With an improved performance, RE-CBRP and MC-CBRP can route much more data packet than DSR and CBRP within a given time.

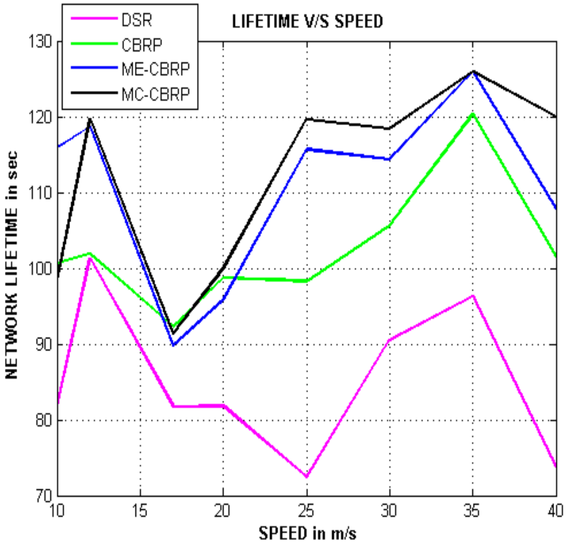


Fig. 1. Network Lifetimes at Different Node Speed

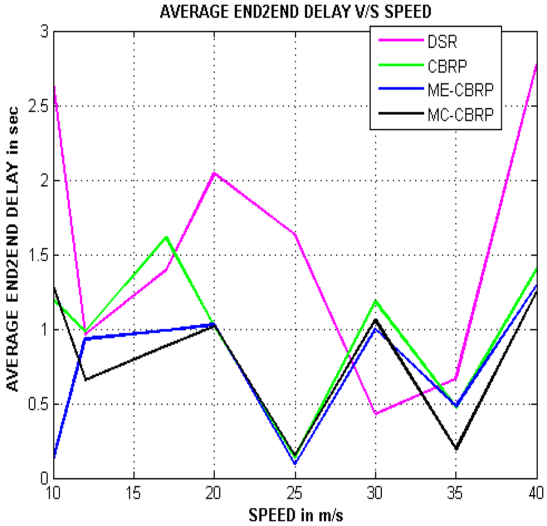


Fig. 2. Average End-to-End Delays at Different Node Speed

Figure 3 shows the throughput (in Kbps) offered by the protocols at various node speeds. Although RE-CBRP and MC-CBRP do not provide extravagant improvement in throughput as compared to DSR and CBRP, their throughput performances are closely comparable. Thus, both the proposed strategies can achieve a data transfer rate compared to DSR and CBRP. With improved network lifetime and end-to-end delay performance and a comparable throughput, both the proposed protocols serve the purpose of efficient utilization of the scarce available energy resource in wireless ad-hoc networks.

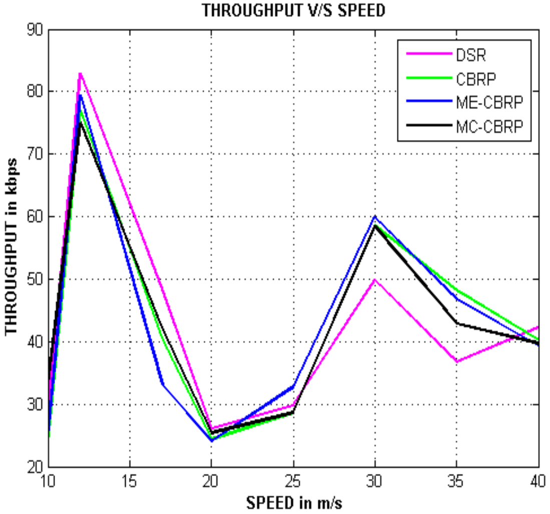


Fig. 3. Throughputs at Different Node

5 Conclusion

The proposed clustering strategies aim to maximize the utilization of nodal energy in an infrastructure less network. Residual Energy CBRP (RE-CBRP) considers only maximum nodal energy whereas Multiple Criterion based CBRP (MC-CBRP) considers nodal energy, node ID and node degree for clustering purposes. Results show that RE-CBRP and MC-CBRP yield elongated network lifetime, shorter end-to-end delay, optimal throughput and less control packets compared to DSR and CBRP with increasing node speed and network traffic. Thus, for clustering based energy efficient routing in a high mobility wireless environment, considering nodal energy as one of the major factor in decision-making, along with other parameters such as node degree and node ID, is crucial. Further work is emphasized on extending the MC-CBRP so as to facilitate energy efficient routing for high mobility and high traffic load environment by exploiting the advantages of multipath routing.

References

1. Tamilarasi, M., Chandramathi, S., Palanivelu, T.G.: Overhead reduction and energy management in DSR for MANETs. In: Proc. IEEE COMSWARE 2008, pp. 762–766 (2008)
2. Nocetti, F.G., Gonzalez, J.S., Stojmenovic, I.: Connectivity based k-hop clustering in Wireless Network. Telecommunication Systems, 205–220 (2003)
3. Gavalas, D., Pantziou, G., Konstantopoulos, C., Mamalis, B.: Clustering of Ad Hoc Networks: An Adaptive Broadcast Period Approach. In: Proc. IEEE ICC, pp. 4034–4039 (2006)
4. Johnson, D.B., Maltz, D.A., Hu, Y.-C.: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). IETF Draft (2007)
5. Johnson, D.B., Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks. Mobile Computing. Kluwer Academic Publishers (1996)
6. Boukerche, A.: Performance Comparison and Analysis of Ad hoc Routing Algorithms. In: Proc. IEEE PCC 2001, pp. 171–178 (2001)
7. Yu, J.Y., Chong, P.H.J., Zhang, M.: Performance of Efficient CBRP in Mobile Ad-Hoc Networks (MANET). IEEE (2008)
8. Jiang, M., Li, J., Tay, Y.C.: Cluster Based Routing Protocol (CBRP). IETF MANET Working Group, Internet-Draft (1999)
9. Yu, J.Y., Chong, P.H.J.: An efficient clustering scheme for large and dense mobile ad hoc networks (MANETs). Computer Communications 30, 5–16 (2006)

Virtual Machine Isolation

A Survey on the Security of Virtual Machines

R. Jithin and Priya Chandran

National Institute of Technology, Calicut, Kerala, India

Abstract. The popularity and widespread adoption of cloud computing has resulted in extensified and intensive use of virtualization technology. Virtualization technology allows the sharing of the same physical resources among several users. This enables the consolidation of servers and a multitude of user machines into a very small set of physical servers, by replacing the physical machines with virtual machines, running on the same physical servers. Consequently, several users work on and store their data in the same physical platform. A software layer is used to enable the sharing of hardware between the different users. Understandably, this leads to apprehensions about the security of their data and working environment for the users, as these are situated only one software layer apart from those belonging to the other users. Centralized storage and centralized computing thus naturally raise the question of security of user's data, and motivate studies on how data security could possibly be compromised. This article surveys the security concerns in virtualization technology. It includes a study of different attacks in the context of virtualization, and logically organizes them in different categories. Where available, the patches to the attacks are also included in the survey. A special focus of the survey is on hardware limitations to support virtualization, and the conclusion drawn is that hardware limitations of different types are the root cause of most of the security issues.

Keywords: Virtualization technology, Virtual Machines, Virtualization Security.

1 Introduction

In the computing scenario, virtualization is the creation of virtual versions of some real objects such as hardware and software. Logical partitions of real objects are made, to create instances of virtual objects. A well-known example is a hard disk drive. Each partition of the hard disk in an operating system is the logical copy of original hard disk.

Two main types of virtualization are *hardware virtualization* and *software virtualization* [1]. *Virtualization software* runs on the real object (i.e., the hardware or software) to be shared. The virtualization software makes multiple virtual objects that look exactly the same real object.

This article focusses on hardware virtualization technology, aimed at partitioning physical machines (computers) into several logical machines. The

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 91–102, 2014.

virtualization software used to create logical machines is popularly termed as *Hypervisor* and each logical machine is referred to as *Virtual Machine (VM)*, in this area. An operating system installed on the virtual machine is known as *Guest Operating System*.

Several practical situations, like surges in the demand for services offered by the virtual machine, or the maintenance of physical servers hosting the virtual machines, may warrant the transfer of a virtual machine from one physical platform to another. This is made possible through the introduction of *Virtual Machine Migration* [2].

Virtual Machine (VM) migration is the process of transferring a virtual machine from one physical machine to another. VM Migration can be done either in active or in passive state of the virtual machine [2]. Migration in *passive mode* is defined as moving a VM from one physical machine to another when the VM is turned off. Migration in *active mode* is defined as moving a VM from some physical hardware to another while the VM is running and without interrupting the services running in VM [2]. Active mode migration is called *Live Migration*. Fig.1 serves to illustrate the process of migration of VMs.

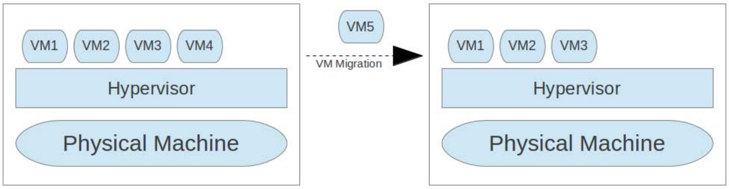


Fig. 1. Virtual Machine Migration Process

Though virtualization technology and technologies realizing the concept of virtual machine migration help to achieve the optimum utilization of physical resources, they spawn several security issues, which are yet to be studied fully and remedied, thus making the system vulnerable to attacks of various kinds. In this paper, various vulnerabilities and attacks in virtualization technology are studied. Generic and specific solutions to these issues, and recent advancements in hardware to overcome these problems are also reported.

The rest of this paper is organized as follows. Section 2 explains the different types of attacks that could exist in virtual machines, as reported in literature. The architectural limitations and the recent advancements in hardware to support efficient virtualization are explained in Section 3. The last section concludes the article with an analysis on the survey.

2 Attacks

From the perspective of the operating system, VMs and physical machines are identical. The perspective of this paper is that the virtual machines should be as secure as physical machines, i. e., there should not be any security vulnerabilities in virtual machines, arising due to virtualization. This implies that virtual machines as isolated logically from each other as are different physical machines, and the only communication mechanism between VMs should be through networking, as in the case of physical machines. This is termed *complete isolation.* Complete isolation is not achieved with existing virtualization technology, as our survey reveals. There are several attacks possible in virtual machines, arising due to the vulnerability introduced by the lack of virtual machine isolation [3] [4] [5] [6] [7] [8] [9] . This paper categorizes them as *Covert Channel Attacks, Malware Attacks* and *Attacks in Migration.* The rest of this section reports on each of these categories of attacks.

2.1 Covert Channel Attacks

Covert channels [3] are the secret channels that exist between two supposedly isolated environments, such as VMs. The existence of covert channels reveals cracks in the isolation, the basic security paradigm, of virtual machines.

Xiao et al. report the construction of a covert channel between virtual machines by exploiting the memory de-duplication feature in the virtualization software i.e., the hypervisor [3]. Virtual machines communicate with the hypervisor through hypercalls, and the hypervisor manages these calls through an event table. *Event-channel* hypercalls and *grant table* hypercalls have been exploited to create a covert channel in [10]. The hypercalls were used to create a shared memory to communicate between two virtual machines. This is a straight covert channel between two virtual machines. Moreover, in [4], a hidden covert channel named *Shared Memory Covert Timing Channel* (SMCTC) was constructed inside such a shared channel by fixing *read* and *write* time intervals. It can be observed that these two covert channels leveraged memory vulnerabilities.

The cache subsystem also has its share of vulnerabilities, open for exploiters. In [4], a covert channel is demonstrated by using the L1 instruction cache as the channel for covert communication between virtual machines. The disaster potential of this covert channel is further emphasized by using the channel to construe an attack to extract the ElGamal decryption key [11] from a victim virtual machine[4]. L2 caches can also be exploited, as introduced in [12], and studied further in several works. [12] quantified the L2 cache based covert channels and assessed the damage potential of L2 cache based covert channels.

Even the silent work horse, the CPU, can become the carrier of covert information. A CPU based Covert Channel between VMs (CCCV) was created by using the CPU work load as the medium of covert communication [13], to work on a single-core processor. Covert Channels using Core-alternation (CCCA) have been demonstrated on virtual platforms with multicore processors [6].

Network resources were also not spared, and have been leveraged to create covert channels between virtual machine. The FCFS packet scheduling system is used as the covert medium between virtual machines in [14]. [15] shows another covert channel, which uses the IP identification field of the IP datagram header as the cover medium, thus exploiting network protocol features to provide covert channels.

Strong isolation between virtual machines at various levels of computer architecture is the need of the day. to prevent covert channels. A model was proposed in [16], for improving isolation of virtual machines. Much research is required into the potential covert channels and integrated approaches to complete isolation of VMs.

The next category of attacks is through malware, which are malicious programs carrying out illegal and unwarranted activities in the system. Though malware are a general threat to computer systems, there are special malware designed exclusively for VMs. They exploit features of VMs or the virtualization environments and technology to carry out their damaging activities, and hence affect only the virtual machines and applications running on them. In the next sub section, two types of VM based malware, named as *VM-aware malware*, and *Hypervisor level rootkits*, are described.

2.2 Malware Attacks

VM-aware malware can identify whether they are running on a virtual machine or real machine [7]. The malware detects the presence of a virtual environment using counter based detection[7] and is possible only on processors with two or more cores. [7] introduces a technique to prevent counter-based detection attack using the Inter Processor Interrupt (IPI) signal.

Virtual Machine Based Rootkit (VMBR) [8] [17] is a hypervisor level rootkit which mimics the structure of a hypervisor. The malware gets installed above the hardware as a hypervisor and the existing operating system is moved to a virtual machine in the hypervisor. The malware works from this hypervisor without its presence getting detected by the operating system, as it runs on a virtual machine running on the spurious hypervisor. VMBR is also called hyperjacking.

Subvirt [8] is a rootkit developed jointly by Microsoft and University of Michigan researchers as an academic example of virtual machine based rootkit. Blue pill [17] is malware based on Intel's x86 virtualization and requires Intel VT-x or AMD-v virtualization support [17]. [18] describes the source code for presenting a minimal hypervisor framework for a rootkit.

Hypervisor level rootkit attack are rendered feasible because systems with virtualization enabled CPUs, when used in the the absence of hypervisors, have just the normal operating systems running above the virtualization enabled CPU [19], and the virtualization capability is leveraged by the pseudo-hypervisors. Gaurdhype [20] is a proposed solution to this problem. Gaurdhype is a lightweight hypervisor which monitors the virtual machine control structures (VMCS)[19]. VMCS is the central part of hardware-assisted virtualization architecture. VMCS contains the states of each virtual machine and hypervisor. By monitoring the VMCS values, the presence of VMBR is determined.

A new approach named *Ether* to analyze the malware in a virtual environment is given in [21]. The idea is to use Intel Hardware Virtualization technology extensions [22] [21]. Due to the architectural limitations in Intel hardware virtualization technology, the malware analysis done by Intrusion Detection Software, Intrusion Prevension Software etc can be detected by the malware itself. It is claimed that these limitations do not exist in AMD hardware virtualization technology [21].

Malware can also help the attacker to create unauthorized access to virtual machines, which violate the isolation property of virtual machines. Malware has been a threat to the operating systems running in physical machines and now prove to be a threat to the operating systems running in virtual machines also. Finding generic and efficient solutions to prevent malware attacks is also a very important research issue.

The third category of attacks is based on a exploiting powerful feature provided by virtualization technology, namely virtual machine migration, which allows virtual machines to be moved from one physical machine to another for logistic or similar reasons. The following subsection reviews various security issues in virtual machine migration.

2.3 Attacks in Migration

The starting point for several attacks in the virtualization environment is the detection of the virtual machine migration process. Detection of the virtual machine live migration has been demonstrated in [23] using ICMP packets. [24] provides a comprehensive survey of vulnerabilities leading to attacks in Live Migration. They are shortly listed here as:

Inappropriate access control policies. Due to inappropriate access control policies any virtual machine can initiate migration, terminate the migration and become susceptible to man-in-the-middle attacks. The attacker can utilize these loop holes in access control to migrate the malicious VM to a hypervisor. The malicious VM in a hypervisor can obviously harm the hypervisor and other VMs [24].

Unprotected transmission channel. Unprotected transmission channel, between the guest and host physical machines involved in the migration process helps the attacker to do passive and active attacks [24].

Loopholes in migration module. Loopholes in migration module, like stack-overflow, heap-overflow and integer-overflow make the migration further vulnerable [24].

Oberheide et al. developed a tool named *Xensploit* [9] to carry out man-in-the-middle attacks on virtual machine migration. Xensploit was used to modify the memory segment, specifically the *sshd* memory segment, in such a way that the *sshd* authentication was be bypassed.

Solutions for preventing the attacks in live migration have been suggested in [24] for secure live migration. These are

Virtual Local Area Network. Virtual Local Area Networks (VLAN) have been proposed for secure migration traffic. VLAN is an invisible network created inside a public Network. VLAN is independent of physical location and is created by tagging the packets with the tag-ID of corresponding VLAN[24].

Network Security Engine. Network Security Engine is a security module proposed to be included inside the hypervisor, and contains protection mechanisms like firewall, IDS and IPS. The goal of the network security engine is to prevent intrusions to the virtual network from outside [24].

Role Based Migration. In a role based migration process, a role based technique using the Trusted Platform Module hardware to find a cryptographically trusted remote hypervisor is used for secure migration. Role based migration process helps to establish policies for deciding who can start migration and to which host and so on[24].

Trusted platform module(TPM) functionality can be leveraged in several other ways as well for secure virtual machine migration. TPM helps to identify the presence of unauthorized access to the system. [25] created a software module named vTPM inside the hypervisor, to share the TPM functionalities with the OS running in each virtual machine. For each virtual machine, an instance of TPM module (vTPM) is created. However, [26] points out that as this implementation is completely inside the software, it cannot protect the cryptographic secrets in every operating system. Due to these limitations Stumpf [26] suggests a different scheme to share the TPM device with the virtual machines.

The observation from our survey on live migration security is that there are several authentication issues, as well as passive and active attacks, that exploit virtual machine live migration. It happens mainly due to the lack of a secure live migration protocol. A secure live migration protocol should provide the following essential security features to a VM Migration process.

- Protected transmission channels
- Integrity of migration data
- Entity authentication

Existing literature reveals only a few secure live migration protocol proposals [26] [27] for the virtual machine migration service. Hence it is clear that researchers on virtualization security should to give more attention to live migration security, as it is a crucial issue.

The above survey points out that the closeness of the hypervisors to the hardware, and the placement of the virtual machine operating systems at a higher software layer is a primary reason for the failure of conventional security mechanisms and existence of hypervisor based malware. Hardware oriented or hardware level solutions seem intuitively possible for securing virtual machines. Hence our survey zooms over to literature on vulnerabilities in virtual machines at various hardware levels. The following section reviews some limitations in different hardware to support virtualization.

3 Architectural Limitations and Consequences

It was seen in the above sections that covert channel attacks exploit the vulnerabilities in CPU, memory and network. Malware uses the vulnerabilities in memory to attack the system. Network vulnerabilities are leveraged by the attacks in VM migration. Literature on hardware devices and features like the CPU, memory subsystem and networks was studied to find the reason for vulnerabilities leading to the mentioned security issues. Architectural limitations in different hardware to support hardware virtualization are the topic of focus in the succeeding subsections.

3.1 Limitations in the CPU Ring Level [28]

In an Intel or AMD processor, hardware virtualization technology facilitates the sharing of the x86 processor among multiple virtual machines. Virtualization technology in Intel or AMD is generally known as x86 virtualization. [1] reports the existence of several vulnerabilities and hardware limitations in x86 virtualization.

IA-32 architecture provides hardware level protection using a 2-bit privilege level called *CPU Ring levels* [28] or *Privilege levels* [1]. Ring level zero for the most privileged instructions, and three for the least privileged instructions. In the non-virtualized environment, an operating system runs its instruction in privilege level 0 and applications run in privilege level 3. In a virtualized environment, the hypervisor runs in level 0. So the guest operating system(on a virtual machine) is forced to run in ring levels other than 0. If a guest operating system also runs in privilege level 0, the guest OS could control the resources which are to be controlled by the hypervisor. The ambiguities that arise in ring level assignments create security threats like *Ring De-privileging, Ring Aliasing, Address Space Compression, Ring Compression* [28].

These security challenges in x86 hardware virtualization have been solved using *hardware-assisted* virtualization technology. Intel and AMD added some features to the existing x86 hardware to solve these security issues. The x86 hardware virtualization technology augmented with these features (from Intel or AMD) is known as Hardware-assisted virtualization technology. Hardware-assisted virtualization technology from AMD is known as AMD-V [29], and that from Intel is known as Intel VT [22].

Virtualization of CPU calls for the execution of system-level and user-level instructions from the guest operating systems. This can be achieved by virtualizing the Instruction Set Architecture (ISA). The next subsection reviews the hardware limitations in ISA virtualization.

3.2 Limitations in ISA Virtualization

In a normal environment an operating system accesses the CPU through the interface named *instruction set architecture.* In a virtual environment, the guest operating system accomplish this through a virtual ISA. ISA virtualization is the technique used in virtualization technology to share CPU, through virtual ISAs.

In 1972, Goldberg stated the sufficient conditions for efficient ISA virtualization [1]. Considering the execution style, he arranged the instructions into 3 classes.

Non privileged instructions. These instructions run in the user mode and can be executed directly in any mode.

Privileged instructions. These instructions trap when the machine is in the user mode and do not trap when machine is in the system mode. The privileged instructions should always trap from user mode.

Sensitive instructions. These instructions can change the system configuration. They are of two types, namely, *control sensitive* and *behavior sensitive* instructions. Control sensitive instructions can change the configurations of system resources, and behavior sensitive instructions produce the result depending on the current configuration of resources.

Goldberg showed formally that for efficient virtualization of ISA, a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions, which implies that, for efficient virtualization, all sensitive instructions should trap in user mode [1].

Satisfying the Goldberg conditions in ISA virtualization will allow the virtual machines to execute non-privileged instructions directly on the hardware, and thereby improve the VM performance [1]. All other privileged and sensitive instructions will move the control of CPU from virtual machine to hypervisor, enforcing VM isolation [1]. Satisfying Goldberg conditions simultaneously achieves isolation and performance for virtual machines at CPU level.

In the year 2000, John Scott Robin et al. [30] analyzed the Intel Pentium processor's ability to support secure virtual machine monitor. His analysis shows that out of 250 instructions in x86 architecture, there are 17 instructions which did not meet the requirement of Goldberg condition of efficient virtualization. It was because these 17 instructions are sensitive instructions but not privileged and they can be executed from the user mode. Smith [1] named these instructions as *Critical Instructions*. Critical instruction are sensitive instructions but not privileged. According to Goldberg, for efficient virtualization there should not be any critical instructions.

Our survey reveals that even a study of existing articles, several books and architecture software developer's manuals related to hardware-assisted virtualization technology [22] [29] [28] [31] does not clarify whether critical instructions exist in the discussed technology. Thus, a detailed practical analysis on the Intel and AMD processor architecture with latest hardware-assisted virtualization technology have to be done to confirm the existence of critical instructions.

In a physical machine, any hardware other than the CPU is considered as an input/output device (I/O device). Virtualization of I/O devices is the next stage of virtualization. Accomplishing I/O virtualization requires overcoming some hardware limitations, as explained in the next sub section.

3.3 Limitations in I/O Virtualization

I/O virtualization is the technique used to share the I/O devices, like storage devices or memory devices among virtual machines. Improper sharing of IO devices (inefficient I/O virtualization) may introduce security vulnerabilities like covert channels [3]. Efficient I/O virtualization is required to improve security of virtual machines. The requirements of I/O virtualization are *Platform independence to operating systems*, *Isolation of I/O devices* and *High Performance that is equivalent to a non-virtualized environment.*

Earlier I/O virtualization was implemented through two techniques named *emulation* [1] and *paravirtualization* [20]. The main advantage of emulation was that it supported a wide range of unmodified guest operating systems. But emulation requires additional transactions between hypervisor and guest OS (Virtual Machine), which increases the complexity of hypervisor, resulting in lower performance [20]. Paravirtualization shows an improved performance over emulation due to reduced interaction between the hypervisor and the guest OS [20]. Paravirtualization requires modification to guest OS. This is a major drawback of paravirtualization. The number of modified operating systems is small yet.

The security requirement *isolation of I/O devices* implies that I/O devices allocated to a virtual machine should not be allocated to or accessed by any other virtual machines at any cost until unallocated. Emulation and Paravirtualization cannot satisfy isolation.

Intel satisfied all those requirements with the introduction of a new technology named Intel VT-d [20]. Intel VT-d is the hardware-assisted virtualization technology for virtualizing the I/O devices. The main design goal of Intel VT-d was to support a wide range of unmodified guest OS with improved security and performance equivalent to that in a non-virtualized environment. Intel VT-d is the major technology of Intel Hardware Virtualization Technology suite, which eliminates various security challenges in I/O virtualization and provides platform independence to operating systems [22].

Intel VT-d architecture is a generalized IOMMU architecture that provides the system software with multiple direct memory access(DMA) protection domains [20]. Intel VT-d provides an improved performance through the direct assignment of I/O devices to virtual machines. Direct assignment is possible through direct memory access technique and provides isolation by mapping I/O devices to a protection domain [20]. Protection domain is a subset of physical memory allotted to a Virtual Machine. One or more I/O devices are allotted to a protection domain.

Our survey reveals that paravirtualization outperformed the initial release of hardware-assisted full virtualization for workloads that perform input/output operations, creating processes, or switch contexts rapidly [32]. We also learnt that IOMMU does not support the virtual machine live migration [33]. These are the major drawbacks of hardware-assisted virtualization technologies. Hence much improvement in hardware assisted virtualization technology is needed through active research, to make it adaptable in general.

4 Conclusion

In this paper, the security issues arising from hardware virtualization, specifically virtualization technology enabling the co-existence of several virtual machines the same physical platform, have been studied, analyzed and reported.

The *attacks* on virtual machines, which potentially result in the compromise of the security of the virtual machine user's data, have been classified into three types, namely the *covert channel* attacks, *virtual machine migration* attacks and *malware* attacks. The three types of attacks leverage the CPU, memory and network respectively, to attack the virtual machines, and thereby violate the isolation property of virtual machines at different hardware levels. The study also includes hardware limitations in CPU and I/O devices to extend support for isolated virtual machines.

It is concluded from the survey that several security issues and hardware limitations exist at the different architectural levels of virtualization technology that compromise the isolation of virtual machines. Only a few of the security issues and hardware limitations have proposed solutions in literature. Covert channels [3], Live migration attacks [33], and the presence of critical Instructions [1] are some of the unresolved issues.

It is inferred from our analysis that each architectural level problem has a general solution. For example, ring level issues in processor can be solved with efficient virtual ring level creation and the critical instruction issue can be solved with efficient ISA virtualization. The implication is that every processor level issue can be solved with efficient processor virtualization, and similarly every memory level security issue can be solved with protected virtual memory, and the network level security issues can be solved with a secure migration protocol. To summarize, our inference is that every security issue can be solved by providing the required solution at the corresponding architectural level. An overview of the solution requirements for assuring the security of virtualization is summarized in Table 1. We conclude that truly isolated virtual machines can be created by completely providing the required solutions, at different architectural levels. This is an area that requires the urgent and devoted attention of researchers in the computing arena, as the proliferation of virtualization technology without sufficient security guarantees can lead to highly vulnerable situations for the users of virtual machines.

Table 1. Security of Virtualization enabled Architecture

Architecture	Security Issues	Required Solution
Processor	- Ring level Problems - Critical Instructions - CPU based Covert channels - VM-aware Malware	Efficient CPU Virtualization
Memory	- Malware Attacks - Memory Covert Channels	Protected Virtual Memory
Network	- Live Migration Attacks - Network Covert Channels	Secure Protocols

References

[1] Smith, J.E., Nair, R.: Virtual Machines: Versatile Platform for Systems and Processes. Morgan Kaufmann (2006)

[2] Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I., Warfield, A.: Live migration of virtual machines. In: Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation, vol. 2, pp. 273–286. USENIX Association (2005)

[3] Xiao, J., Xu, Z., Huang, H., Wang, H.: A covert channel construction in a virtualized environment. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS 2012, pp. 1040–1042. ACM, New York (2012), http://doi.acm.org/10.1145/2382196.2382318

[4] Wu, J.Z., Ding, L., Wang, Y., Han, W.: Identification and evaluation of sharing memory covert timing channel in xen virtual machines. In: 2011 IEEE International Conference on Cloud Computing (CLOUD), pp. 283–291. IEEE (2011)

[5] Zhang, Y., Juels, A., Reiter, M.K., Ristenpart, T.: Cross-vm side channels and their use to extract private keys. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS 2012, pp. 305–316. ACM, New York (2012), http://doi.acm.org/10.1145/2382196.2382230

[6] Li, Y., Shen, Q., Zhang, C., Sun, P., Chen, Y., Qing, S.: A covert channel using core alternation. In: 2012 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 324–328. IEEE (2012)

[7] Li, H., Zhu, J., Zhou, T., Wang, Q.: A new mechanism for preventing hvm-aware malware. In: 2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN), pp. 163–167. IEEE (2011)

[8] King, S.T., Chen, P.M.: Subvirt: Implementing malware with virtual machines. In: 2006 IEEE Symposium on Security and Privacy, p. 14. IEEE (2006)

[9] Oberheide, J., Cooke, E., Jahanian, F.: Empirical exploitation of live virtual machine migration. In: Proc. of BlackHat DC convention (2008)

[10] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. ACM SIGOPS Operating Systems Review 37(5), 164–177 (2003)

[11] ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory 31(4), 469–472 (1985)

[12] Xu, Y., Bailey, M., Jahanian, F., Joshi, K., Hiltunen, M., Schlichting, R.: An exploration of l2 cache covert channels in virtualized environments. In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, CCSW 2011, pp. 29–40. ACM, New York (2011), http://doi.acm.org/10.1145/2046660.2046670

[13] Okamura, K., Oyama, Y.: Load-based covert channels between xen virtual machines. In: Proceedings of the 2010 ACM Symposium on Applied Computing, SAC 2010, pp. 173–180. ACM, New York (2010), http://doi.acm.org/10.1145/1774088.1774125

[14] Gong, X., Kiyavash, N., Venkitasubramaniam, P.: Information theoretic analysis of side channel information leakage in fcfs schedulers. In: 2011 IEEE International Symposium on Information Theory Proceedings (ISIT), pp. 1255–1259. IEEE (2011)

[15] Jaskolka, J., Khedri, R.: Exploring covert channels. In: 2011 44th Hawaii International Conference on System Sciences (HICSS), pp. 1–10. IEEE (2011)

[16] Kaleeswaran, S.: Managing covert information leaks in xen virtual machine systems. Master's thesis. NIT Calicut (May 2010)

[17] http://en.wikipedia.org/wiki/Blue_Pill_%28software%29

[18] Myers, M., Youndt, S.: An introduction to hardware-assisted virtual machine (hvm) rootkits. White Paper of Crucial Security (2007)

[19] Carbone, M., Lee, W., Zamboni, D.: Taming virtualization. IEEE Security & Privacy 6(1), 65–67 (2008)

[20] Abramson, D., Jackson, J., Muthrasanallur, S., Neiger, G., Regnier, G., Sankaran, R., Schoinas, I., Uhlig, R., Vembu, B., Wiegert, J.: Intel virtualization technology for directed i/o. Intel Technology Journal 10, 178–192 (2006)

[21] Dinaburg, A., Royal, P., Sharif, M., Lee, W.: Ether: malware analysis via hardware virtualization extensions. In: Proceedings of the 15th ACM Conference on Computer and Communications Security, pp. 51–62. ACM (2008)

[22] Hardware-assisted virtualization technology, Intel, Web Article (2012), http://www.intel.in/content/www/in/en/virtualization/virtualization-technology/hardware-assist-virtualization-embedded-technology.html (retrieved December 10, 2012)

[23] König, A., Steinmetz, R.: Detecting migration of virtual machines. In: Proceedings of the 10th Würzburg Workshop on IP: Joint ITG, ITC, and Euro-NF Workshop Visions of Future Generation Networks (EuroView 2011), Julius-Maximilians-Universität Würzburg, Lehrstuhl für Informatik III (2011)

[24] Shetty, J., Anala, M.R., Shobha, G.: A survey on techniques of secure live migration of virtual machine. International Journal of Computer Applications 39(12) (2012)

[25] Perez, R., Sailer, R., van Doorn, L.: vtpm: virtualizing the trusted platform module. In: Proc. 15th Conf. on USENIX Security Symposium, pp. 305–320 (2006)

[26] Stumpf, F., Eckert, C.: Enhancing trusted platform modules with hardware-based virtualization techniques. In: Second International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2008, pp. 1–9. IEEE (2008)

[27] Danev, B., Masti, R., Karame, G., Capkun, S.: Enabling secure vm-vtpm migration in private clouds. In: Proceedings of the 27th Annual Computer Security Applications Conference, pp. 187–196. ACM (2011)

[28] Neiger, G., Santoni, A., Leung, F., Dion Rodgers, R.U.: Intel virtualization technology: Hardware support for efficient processor virtualization. Intel Technology Journal 10, 166–177 (2006)

[29] Advanced Micro Devices: Secure Virtual Machine Architecture Reference Manual Advanced Micro Devices (May 2005)

[30] Robin, J.S., Irvine, C.E.: Analysis of the intel pentium's ability to support a secure virtual machine monitor. DTIC Document, Tech. Rep. (2000)

[31] Uhlig, R., Neiger, G., Rodgers, D., Santoni, A.L., Martins, F.C., Anderson, A.V., Bennett, S.M., Kgi, A., Leung, F.H., Smith, L.: Intel virtualization technology, May 2005, pp. 48–56. IEEE Computer Society Press (2005)

[32] Adams, K., Agesen, O.: A comparison of software and hardware techniques for x86 virtualization. ACM SIGOPS Operating Systems Review 40(5), 2–13 (2006)

[33] Ben-Yehuda, M., Mason, J., Xenidis, J., Krieger, O., Van Doorn, L., Nakajima, J., Mallick, A., Wahlig, E.: Utilizing iommus for virtualization in linux and xen. In: OLS 2006: The 2006 Ottawa Linux Symposium, pp. 71–86. Citeseer (2006)

Trust Based VM Consolidation in Cloud Data Centers

T.R. Venugopal Anandharajan and Marcharla Anjaneyulu Bhagyaveni

Department of ECE, College of Engineering, Guindy, Anna University,
Chennai – 25, India
anandharajan.t.r.v@gmail.com, bhagya@annauniv.edu

Abstract. Virtualization in Cloud data center, handle workloads and maintain SLA providing a better QoS to the Cloud consumer will lead to the harnessing of the present Cloud Computing infrastructure. Our model is on a statistical property and based on reliability and reputation combined for a “trust” based that we design our algorithms to handle QoS and these algorithms prove better than the existing model. However, the growing demand of the resources (physical) in a data center has drastically increased the energy consumption of computations (cyber) being processed in data centers, which has become a decisive issue. To address the trade-off between performance and power consumption we propose a near-optimal scheduling policy based on the CQR (Composite Quantile Regression) and the Minimum energy heuristics (MPP) to find a trust based Cloud character probability modeling that exploits heterogeneity across multiple data centers for a Cloud provider.

Keywords: Cyber Physical Systems, Trust Management, Load Balancing, Green IT, SLA analysis.

1 Introduction

The responsibility for IT assets and the maintenance of those assets is shifted to the Cloud service provider. In April 2007, Gartner estimated that the Information and Communication Technologies (ICT) industry generates about 2% of the total global CO_2 emissions [1] which are equal to the aviation industry. Like the automobile industry the government may impose restriction to the emissions from a data center. Thus, meeting the Cloud demand the CO_2 efficient data centers are to be readymade. Environmentally sustainable data centers are necessary to reduce the environmental impact to reduce a small percentile of the global warming. Future scalability of Cloud may be achieved with better efficiency and environmental friendly systems. Users of Cloud services are interested on value received from the Cloud than its mechanics. The Cloud data centers inefficient use of the resources has been the major reason for the Cloud energy consumptions even when there is good availability of computational resources and efficient hardware in Data centers. Over a six month period the data collected from 5000 servers around the world shows that only 10-50% of the servers has been used efficiently [2]. This has led to the capital expenditure over the data centers and the running expenditure like maintain monitoring of the over-provisioned

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 103–114, 2014.

resources. Underutilized servers consume 70% of the power which adds to the running expenditure [3].

2 Related Work

Imada et. al [4] investigates power along with QoS (Quality of Service) overall performance characteristics of virtualized hosts with virtual machine technological know-how. The virtualized servers have to handle the plenty of the VMs through the provider. The other works [5], [6], [7] have considered specific component and an exhaustive QoS research were experimented. These papers did not consider VM consolidation and only alive status of the hosts which is a binary property was considered. From the literature challenges, we addressed a statistical trust based energy harnessing in this paper. The trust based environment is the one which would handle the VMs in a data center to maintain equilibrium or coalesce between power and performance of the available resource for all probable nodes. Most literature tried to handle this trade-off but on different strategies. VM consolidation is an important solution to the trade-off.

Table 1. Literature Survey in QoS perspective in Cloud IaaS

Work Citation	Reliability taken into account?	If QoS taken into account, Wat was the basic component on which it was implemented?	QoS analysis in Datacenter?
[4]	Yes	Statistical Analysis of CPU Characteristics	Yes
[5]	No	CPU Charactersitics	Yes
[6]	Yes	VM Charactersitics	Yes
[7]	Yes	CPU Charactersitics	Yes

From the above literature survey we were able to arrive at a conclusion where VM consolidation and component consideration where necessary for a trust model. We have implemented some VM consolidation algorithms in our previous papers [8, 9]. In this paper we have concentrated on the methods as mentioned below

1. The VM consolidation and the Host oversubscription detection was necessary to evaluate an efficient data center. The CQR technique based Host overloading detection was adapted and we obtained efficient results.
2. The Host status analysis in various literatures was not dealt with so we proposed a Cloud Character Model where host trust based characteristics' were studied based on the proposed EIRP (Energy / Instruction Rate Performance) based Energy model.
3. To determine the repute we defined metrics on a trust perspective. Since Energy Consumption was the main consideration we analysed the various Host Oversubscription and VM selection algorithms based on the legacy algorithms.

3 CQR Based MPP Policy

VM selection and Host Oversubscription techniques have been discussed in this section. MPP (Minimum Processing Power) policy has been proposed and VMs are consolidated. CQR (Composite Quantile Regression) has been adapted to analyse the Host Oversubscription and the oversubscription is handled based on our proposed EIRP metric and analysed.

3.1 Minimum Processing Power Policy

The Minimum Processing Power (MPP) policy proposes a host where a VM from $v = \{v_1, \ldots, v_n\}$ that requires the minimum processing power to complete a migration relatively to the other VMs allocated to the available host based on our proposed MEP algorithm to the available host. The minimum processing power is estimated as the amount of power utilized by the VM divided by the MIPS available for the host j as given in equation 3. Let V_k be a set of VMs currently allocated to the host. The energy curve is usually non-linear. We define the MPP parameter Energy per Instruction Rate Performance (EIRP) for a host having VMs as

$$EIRP_{Host} = \frac{Power_h}{MIPS_h} \tag{1}$$

$$v \in V_h \mid \forall\, a \in V_h, \frac{Power_u\,(requesting)}{MIPS_h} \le \frac{Power_u\,(processing)}{MIPS_h} \tag{2}$$

where V_h be a set of VM currently allocated to host h and a is the currently processing VM in a host at an instant of time.

3.2 CQR Host Overloading Detection

For the MPP policy we combine the CQR technique instead of Local Regression and predict better based on the upper utilization of the CPU utilization. In the A.Beloglazov and R. Buyya [7] paper they have judged the LR MMT is the efficient algorithm where the loss function could be approximated to give an efficient algorithm. We base our algorithm on the Composite Quantile Regression proposed by Hui and Yuan [10] which can estimate the regression coefficients in the classical linear regression model. The relative efficiency of the CQR estimator compared with the least squares estimator is greater than 70% regardless of the error distribution. Furthermore, the CQR estimator could be much more efficient and sometimes arbitrarily more efficient than the least squares estimator. The main idea of the method of local regression is fitting simple models to localized subsets of data to build up a curve that approximates the original data. From the workload CPU utilization instances, we form localized subsets and implement CQR to find a better way to handle host oversubscription if it takes place.

If $(t_i - y_i)$, i=1….n is an independent and identically distributed random sample of the CPU utilization taken from the PlanetLab trace. We estimate $x(t) \approx x(t_0) +$

$x^{'}(t - t_0)$ and then fit a linear model locally in the neighbourhood of t0. Let k(·) be a smooth kernel function, the local linear regression estimate in x(t0) in â, when

$$(\hat{a},\hat{b}) = arg_{a,b}^{min}[\sum_{i=1}^{n}\{y_i - a - b(t_i - t_0)\}^2 \, k(\tfrac{t_i - t_0}{h})] \tag{3}$$

Where h is the smoothing parameter.

The best properties that CQR combines are the design adaptation property and high mini-max efficiency property. This method helps in finding the host oversubscription and involves the constraint as to analyse based on the response time and the CPU operating frequency.

The observations are estimated for a cost function which gives the predicted trend line of the CPU utilization from the PlanetLab trace on the right . We propose the Composite Quantile Regression as follows.

Let

$$\varphi_{l_m}(r) = l_m a - aI(r < 0), m = 1,2, \dots, q \tag{4}$$

For q check loss functions at q quantile positions

$$l_m = \frac{m}{q+1} \tag{5}$$

The size of the subset is defined by a parameter of the method called bandwidth. In the linear regression model the CQR loss is defined as

$$\sum_{l=1}^{q} \sum_{i=1}^{n} a^{n-k} \, \varphi_{l_m}(y_i - a_m - \, bt_i) \tag{6}$$

CQR combines the strength across multiple quantile regressions with forcing a single parameter for the slope. In our CQR algorithm, for each new observation from $\widehat{a_1}, \widehat{a_2}, \widehat{a_3}, \dots \dots, \widehat{a_q}, \hat{b}$ we find a new trend line.

$$\widehat{a_1}, \widehat{a_2}, \widehat{a_3}, \dots \dots, \widehat{a_q}, \hat{b}^{CQR} = \underset{a_1,a_2,\dots,a_q,b}{arg\,min} \sum_{l=1}^{q} \sum_{i=1}^{n} a^{n-k} \, \varphi_{l_m}(y_i - a_m - \, bx_i^T) \tag{7}$$

The trade-off has to be addressed based on the QoS that can be satisfied when the SLAV can be efficiently maintained or minimized. The host (or) processor available can have different status based on their availability in any point in time. The execution time and time analysis of various parameters depend on the CPU operating frequency. The local linear estimates the VM host oversubscription precisely and efficiently than the Local Regression.

4 Cloud Character (Trust) Model

In this section we propose the Cloud Character model to analyse the above mentioned algorithms based on trust perspective. Data center updates the information of its resources and they are hard estimate runtime of parallel applications where execution time is inversely proportional to CPU operating frequency. These trust reliable ratings are learned over time based on the results returned by the provider to the server.

Cloud data center servers' alive status is analysed as a binary property in most of data center literature. We analysed a different dimension where instead of a binary property we propose a statistical distribution for the data center analysis. The inherent unreliability nature of distributed systems was the major motivation behind this proposal. All Cloud providers report their results to a centralized server, so a local statistical character based reputation system could be employed. We estimate a provider's repute $\mathcal{R}$ at a time t as follows

Table 2. Statistical Distribution to model information on Host Status

Parameters	Distribution (0,1)	Field Density (Host)
General (G)	Uniform	Regular Cloud traffic
Dependable (D)	1-Pareto(a=1, b=0.1)	More free, Few busy
Less dependable (LD)	Pareto(a=1, b=0.2)	More busy, More accidents
Appropriate (A)	Normal: μ=0.9, σ=0.05	Almost free
Equally probable (E)	Bi-Normal: μ=0.2/0.8, σ=0.1	50% equally free and busy
Less likely (LL)	Normal: μ=0.3, σ=0.1	Less realized(on the fly Cloud)

To analyse the cloud trust for reliability as in Table 2 we define the analytical parameters to be considered and these analytical parameters intend to solve the Trust perspective by giving an awareness of what trust a customer can expect from a cloud prrovider. Coalesce is to find a better trusted environment to allocate VMs to form a balance between power and performance. The definition and assumption for these analyses is as follows

Definition 1: Job: An application i of the form of a tuple which is heterogeneous in a Cloud environment and may be application domain dependent. Independent users submit requests for provisioning of heterogeneous $\mathscr{v}$ VMs characterized by requirements to processing power defined in MIPS, amount of RAM and network bandwidth, the combination of mixed workload from different users lead to mixed workload and can be of various types of applications, such as HPC and web-applications, which utilize the resources simultaneously.

Definition 2: Data center Host Space: The data center host space is the set of the available host lists where VM is free based on our model to provision the jobs.

Definition 3: Repute: The reputation based Cloud character where a provider returns the exact possibility of dependence of a host.

Definition 4: Alive Host Group: The nodes which can handle the jobs at some time.

We now present a Character based scheduling as in Table 2, for handling the VM $\mathscr{v}$ in a host for the efficient processing of the incoming jobs and to avoid underutilization and also avoid oversubscription of the VMs. In addition, we also find the best Cloud environment for the job to acquire its share of processing space and the Coalesced data center which is to attain the objectives

Strike (η): The Strike during an allocation is the number of Host in a data center Host Space for which many VMs were processed successfully providing a better repute in an Alive host group.

$$\eta = host_{VM_s} \tag{8}$$

where $host_{VM_s}$ is the set of VMs successfully completed during that allocation.

Hostware (ζ): The hostware is the mean number of Hosts in a data center Host space assigned to each VM during allocation in an Alive host group.

$$\zeta = \frac{\sum_{i=1}^{v} J}{v} \tag{9}$$

where J is the total number of hosts available during an allocation.

Spare (Θ): The Spare during an allocation is defined as the ratio of successfully completed v VMs to the number of hosts available during the allocation, Considering the Alive host group.

$$\Theta = \frac{\eta}{J} \tag{10}$$

We simulated exhaustive simulations of the Statistical distribution we have considered and accordingly: the Cloud Server Character distributions described in Table 2, we take the host as N heterogeneous physical nodes. For a given distribution and Max_{hosts}^{a}, we set a best possible value equal to the spare Θ. The algorithms defined earlier like the MeP, Ff, GP are all used and they form groups for each of the Cloud Character distribution and the corresponding values are plotted in graphs as shown in the later section.

The allocation of VMs $v = \{v_1, \ldots, v_n\}$ of n, $n \in \mathbb{N}$ to $h = \{h_1, \ldots, h_m\}$ of m, $m \in \mathbb{N}$, hosts. The algorithms defined below randomly assigns hosts to various group of VM in host considered for N heterogeneous physical nodes, maximum available hosts Max_{hosts}^{a} which is the least individual number of under-loaded host in the host list and minimum available hosts Min_{hosts}^{a} which is the least individual number of over-loaded hosts in the hosts list.

Algorithm 1: The Minimum Energy Performance Algorithm (MeP)

```
      Input: hostlist, vmlist            Output: Best e-rated host
to which $v$ VMs are allocated
  Sort $Max_{hosts}^{a}$ vmList.sortDecreasingUtilization()
      foreach vm in vmList do
             minPerformance←MAX
             availableHost←$Min_{hosts}^{a}$
      foreach host in hostlist do
      if isHostUnderloaded(host) then
             performance ←estimatePerformance(host,vm)
      if performance<minPerformance then
             availableHost←host
             minPerformance←Performance
      if availableHost≠$Min_{hosts}^{a}$ then
             available.add(vm,availableHost)
  return available
```

From the above mention MPP policy we analysed the VM selection and Host oversubscription algorithms and implemented the Energy Performance aware algorithms.

Algorithm 2: The Greedy Power Algorithm (GP)

```
Input: hostlist, vmlist Output: Best stored host in hostlist to which 𝓋 VMs are allocated
Sort $Max_{hosts}^{a}$
     vmList.sortDecreasingUtilization()
foreach vm in vmList do
foreach host in hostlist do
     if isHostUnderloaded(host) then
     diff.powerconsumption←estimatePowerconsumption(host,vm)
     if diff.powerconsumption is minimum then
            min diff.powerconsumption←host
     host←add.vmList(vm,host)
endif
endif
return
```

The first fit is going to pair VMs to servers on the basis of a descending and ascending order of utilization at some instant of time and the firsts in each are paired for processing.

Algorithm 3: The First-fit Algorithm (Ff)

```
Input: hostlist, vmlist Output: First stored host in hostlist to which 𝓋 VMs are allocated
Sort $Max_{hosts}^{a}$
     vmList.sortDecreasingUtilization()
foreach vm in vmList do
foreach host in hostlist do
     if isHostUnderloaded(host) then
            hostList.sortIncreasingUtilization()
            host←estimateutilization(host,vm)
     host←add.vmList(vm,host)
endif
return
```

The complexity of the algorithms is $h * v$ number of Host and the number of VM respectively. The Greedy power algorithm sorts the best power aware hosts on the greedy heuristics perspective and the VMs are allocated. The delta increase in the power consumption is calculated by the difference between the time considered for the power consumed and the minimum difference among the hosts in a host list at some point in time is taken into account. Since host has been handling incoming VM's host oversubscription is possible and hence the above policy is used to iterate the following algorithm to avoid the host oversubscription.We use the above three algorithms to form groups by randomly adding hosts until coalesce is met. Performance and energy consumption depends on the availability of efficient resources and scarcity of efficient resources burdens time of SLAV and VM migration.

5 Experiment Setup and Workload Data

Cloud computing creates a view of infinite computing on the Cloud consumers, it is crucial to deploy resource allocation algorithms with a large scale virtualized information center infrastructure. In real infrastructure it is not viable to execute repeated experiments for the proposed algorithms; it is not an economical tool [5]. Therefore, to guarantee the accuracy and repeatability regarding experiments, simulations have been chosen in an effort to evaluate the performance with the proposed heuristics. For the experiments we include chosen CloudSim toolkit [11]. All kinds of other simulation toolkits can be purchased but modelling with the virtualized environments in addition to on-demand resource provisioning is just not available in people toolkits. Energy consumption modelling, dynamic workloads to simulate service applications are already incorporated in this toolkit so that it is an appropriate alternative. Pertaining to simulation level experiments, it is vital to perform experiments using real-world traces from a real system available. We have conducted tests from real-world traces obtained from CoMon monitoring infrastructure [12], the monitoring commercial infrastructure of PlanetLab. During simulations, usually every VM is randomly given a workload search from one of the VMs through corresponding day since the workload is usually on days to weeks CPU utilization by more than of 1000 VMs through 500 places all over the world. The workload traces are in an interval of 300 seconds and for our simulation we have now taken records collected while in March as well as April 2011 [13].

6 Results

The Effect of the VM size with respect to the VM migration and Energy has been analysed for a day at the Planetlab in Fig. 1 where the data center is of 800 heterogeneous nodes half of which are HP Proliant ML 110 G4 servers and other half are HP Proliant ML 110 G5 servers. The server utilization and power consumed by these servers are taken from real data from SPECpower benchmark rather than an analytical model of a server which makes the simulation more efficient. The time of completion of the Virtual Machine Migration helps in the reduction of SLAV and helps in the QoS that can be delivered. The Energy fluctuates due to the resource availability and the performance harnessing helps in better tackling this issue. As the Energy consumed decreases from 100 kWh, the VM migration reduces for our proposed model exponentially by 5% for a certain density of VM.

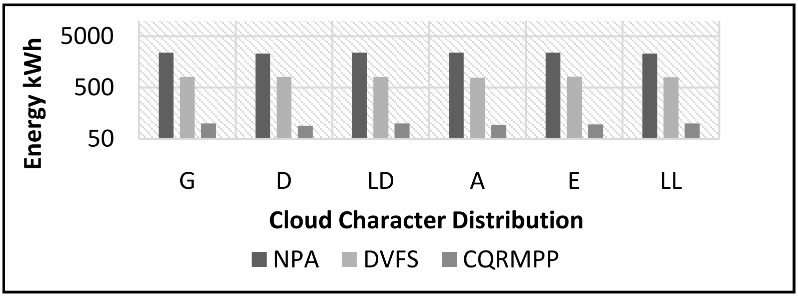


Fig. 1. Energy spent for the Cloud Character Distribution

In Fig. 1 the analysis of the Energy for NPA (Non Power aware) and DVFS (Dynamic Voltage Frequency Scaling) is done with respect to our proposed model which is drastically efficient when compared to NPA. For a G distribution the Energy for NPA is around 2400kWh whereas CQRMPP is 120kWh. Energy Consumption depends on the hosts in a data center and QoS to a customer can be satisfied only when a lesser SLAV occur which in turn depends on the Virtual Machine Migration takes a lesser time and with a lesser frequency of occurrence.

Energy when the Cloud character distribution is compared has an improvement with the proposed CQR where localized subsets are analysed and the algorithms are simulated, CQRMPP which combines the CQR and minimum energy heuristics and thus enables the best energy as in Fig. 1 saving of up to 2000kWh compared to NPA and 40 to 50% lesser energy compared to DVFS. Thus energy is better for proposed system and efficient when compared to the traditional systems. We perform multi-objective optimization and forms coalesce between the consumer and the provider to handle the trade-off between problem of energy and performance efficient dynamic consolidation of VMs performance always hinders the energy perspective. For our simulation experiments as shown in the Fig. 2 the mean strike across multiple simulations conducted the theoretical spare is around 40 to 60% the minimum energy performance and first fit algorithm improve the strike and the Greedy Power is almost good as MeP. The first fit is outperformed by 10 to 20% by the MeP and GP. The Strike and spare for various real time scenario of the Cloud Character model helps to study the available VMs which can be allocated to a host with a larger probability of availability.

We analyse every algorithm and try to improve the strike and spare of coalesce between the processes and processing element. The spare of MeP and GP is better than the Ff for A and LL. First fit mostly deviates and is not as performing as the other two algorithms. The host status varies at a point in time and has to be harnessed to attain better QoS. There can be several unreliable coalesce from the available (or) assumed high reliability character based distribution that we exhibit. Energy consumption in Fig. 3 is lesser to a major extent when MeP is in play. In LD and E the First fit has got at 10% increase in energy and it is due to the distribution assumed and the limit of convergence. Performance Degradation is comparatively lesser for MeP and amounts to 20 to 30% decrease in Performance Degradation in D, G, E where there is better QoS since SLAV will also decrease which in turn helps to solicit the performance to be achieved. The characteristics of the servers and data on their power consumption are considered.

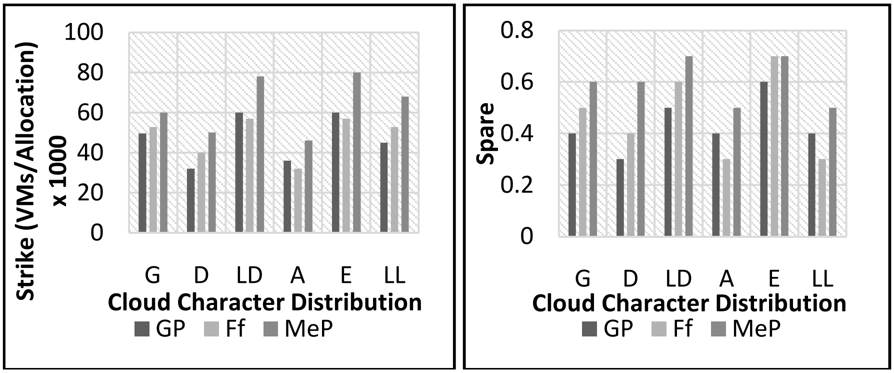


Fig. 2. Strike and Spare for the Cloud character Distribution analysing three different algorithms for repute

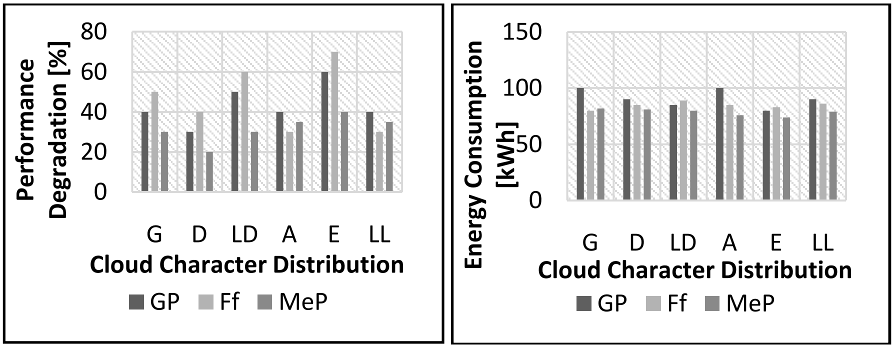


Fig. 3. Performance Degradation and Energy Consumption for the Cloud character Distribution analysing three different algorithms for repute

The metrics we consider is the Performance degradation and the Energy consumption by physical nodes. The reduction of PDM is necessary as energy consumption increases. Our proposed model has proved better when energy consumption decreases the PDM is comparatively better when a traditional algorithm is being executed. The VM use lesser workload as specified in PlanetLab workload data, the consolidation of VM is hence necessary and devising algorithm for this consolidation has been simulated by our model. Our model is efficient in the trade-off point of view with respect to performance and power consumption.

Table 3. Energy spent and the SLATAH on 800 heterogeneous nodes in a 24 hour simulation

No. of VMs	Legacy Model ENERGY	CQRMPP ENERGY	Legacy Model SLATAH	CQRMPP SLATAH
1052	116.71	105.80	19.54%	4.03%
872	100.39	89.87	16.13%	3.68%
772	80.78	71.28	15.65%	3.65%
662	61.46	54.25	15.18%	3.25%
462	41.62	36.94	11.88%	2.77%
262	27.44	24.57	8.58%	2.44%

The SLA performance degradation due to migration for the traditional algorithms is almost zero percentage and as consolidation is done the proposed algorithm harness the resources, increasing the SLATAH but in turn reduces Energy consumption. Energy levels are exponentially 10% lesser throughout the simulation and CQRMPP SLA performance degradation due to migration is almost linear at 0.06% throughout the simulation considered. As the density of VMs increase and Energy consumption increases the CQRMPP SLATAH increases by about 0.05% as in Table 3. The density of VMs when increased shows linear dependence of the number of Host

Shutdown when there is an exponential increase. The energy spent and the CQRMPP SLATAH of the proposed system proved efficient since SLATAH was higher the energy decreased due to better VM consolidation. Thus our proposed algorithms CQRMPP proves to be efficient than the legacy algorithms to about 27%.

The mean value of the sample means of the time before a host is switched to the sleep mode for the CQR-MPP algorithm combination is 864 seconds with the 95% CI: (820, 908). Performance Degradation is higher since there is more utilization of resources under constraints unlike the hosts being overused and more servers left underused or not used at all. This means that on an average a host is switched to the sleep mode after approximately 14.4 minutes of activity. The mean number of host transitions to the sleep mode for our experiment setup (the total number of hosts is 800) per day is 771 with 95% CI: (707, 835). The mean value of the sample means of the time before a VM is migrated from a host for the same algorithm combination is 20.26 seconds with the 95% CI: (19.9, 20.62). The mean value of the sample means of the execution time of the CQR-MPP algorithm on a server with an Intel Core i7 (2.40 GHz) processor and 2 GB of RAM is 0.10 ms with the 95% CI: (0.09, 0.11). The SLATAH and PDM vary as Energy of a simulation having 1052 VMs and 800 heterogeneous nodes the lesser the Energy consumption, higher is the SLAV and this is reflected in SLATAH. Consolidation of the hosts in turn increases the Host Shutdown increasing the Energy requirement of the available hosts and Migration Time lessens which increases SLAV. Consolidation of the hosts in turn increases the Host Shutdown increasing the Energy requirement of the available hosts and Migration Time lessens which increases SLAV.

7 Conclusion

We have deployed the Cloud energy efficient strategies by dynamic consolidation and statistical models to harness the trust that is required for a consumer on Cloud. Thus the trade-off has been harnessed by our proposed model and simulated. The SLA and QoS metrics lead to the trade-off between problem of energy and performance efficient dynamic consolidation of VMs. The results have proved to be better than the available traditional methods in the Energy perspective in the present Cloud infrastructure and maintaining the server utilization to a better level and avoid fluctuation of servers or hosts thereby reducing the running expenditure of the over-provisioned servers. The proposed algorithms prove better and have harnessed the energy consumption although for a when higher energy consumption maintained a lower SLAV thereby decreasing the carbon footprint in the present IT infrastructure.

Acknowledgments. This work was supported by the University Grants Commission (UGC), New Delhi, India. RGN Fellowship Scheme, grant dated November 2010.

References

1. Pettey, C.: Gartner estimates ICT industry accounts for 2 percent of global CO2 emissions (2007)
2. Barroso, L.A., Hölzle, U.: The Case for Energy-Proportional Computing. Computer 40, 33–37 (2007)
3. Fan, X., Weber, W.D., Barroso, L.A.: Power provisioning for a warehouse-sized computer. ACM SIGARCH Computer Architecture News 35, 13 (2007)
4. Imada, T., Sato, M., Kimura, H.: Power and QoS performance characteristics of virtualized servers. In: 2009 10th IEEE/ACM International Conference on Grid Computing, pp. 232–240 (2009)
5. Mei, Y., Liu, L., Pu, X., Sivathanu, S., Dong, X.: Performance analysis of network i/o workloads in virtualized data centers (2011)
6. Gao, Y., Guan, H., Qi, Z., Wang, B., Liu, L.: Quality of Service Aware Power Management for Virtualized Data Centers. Journal of Systems Architecture (2013)
7. Wang, L., Wang, H., Cai, L., Chu, R., Zhang, P., Liu, L.: A Hierarchical Memory Service Mechanism in Server Consolidation Environment. In: 2011 IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS), pp. 40–47 (2011)
8. Anandharajan, T.R.V., Bhargavan, D., Bhagyaveni, M.A.: VM Consolidation Techniques in Cloud Datacenter. Journal of Theoretical and Applied Information Technology 53, 267–273 (2013)
9. Anandharajan, T.R.V., Bhagyaveni, M.A.: Co-operative Scheduled Energy Aware Load-Balancing technique for an Efficient Computational Cloud. International Journal of Computer Science Issues 8, 571–576 (2011)
10. Zou, H., Yuan, M.: Composite quantile regression and the oracle model selection theory. The Annals of Statistics 36, 1108–1126 (2008)
11. Buyya, R., Ranjan, R., Calheiros, R.N.: Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In: International Conference on High Performance Computing & Simulation, HPCS 2009, pp. 1–11 (2009)
12. Park, K.S., Pai, V.S.: CoMon: a mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Operating Systems Review 40, 65–74 (2006)
13. Beloglazov, A., Buyya, R.: Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. Concurrency and Computation: Practice and Experience 24(13), 1397–1420 (2012)

A Trust Model for Directory-Based Service Discovery in Mobile Ad Hoc Networks

R. Deepa and S. Swamynathan

Department of Information Science and Technology, College of Engineering,
Anna University, Chennai, India
rdeepa5@ymail.com, swamyns@annauniv.edu

Abstract. Trust computation and management are significant areas of research in mobile ad hoc networks, since the mobile nodes are vulnerable to security threats. A trust mechanism allows the participating nodes to maintain trust relationships among themselves, by evaluating one another and deciding which nodes can interact with one another and whether they can. In this paper, a trust model for the directory-based service discovery is proposed, using the Dezert-Smarandache Theory (DSmT). The trust value of a node is computed by another node, based on the behavior of nodes, when a service provider node provides the service to the requesting node, during the service discovery process. An evidential theory, DSmT handles uncertain and paradoxical information that may arise from the fusion of several sources of evidences. Simulations were carried out, to analyze trust values of the requester nodes and packet delivery ratio, to show the efficiency of the proposed model.

Keywords: trust, mobile ad hoc networks, service discovery, directory nodes, Dezert-Smarandache Theory, direct trust, recommendation, evidential theory.

1 Introduction

A mobile ad hoc network (MANET) is a network of mobile nodes without any fixed infrastructure. These nodes are connected by wireless links. Since MANETs are deployed in uncontrolled environments, when the nodes communicate with other nodes, their behavior changes with time and the environmental conditions. Thus, trust is important for computing the nodes' behavior to ensure the proper operation of networks [1,2]. Trust is defined as "the degree of subjective belief about the behavior of the particular entity" [3]. The trust can be computed using the theory of evidence. The evidential theory can be used for the partial knowledge, belief updating, and for combination of evidence [4].

The objective of this paper is to create, and manage trust in MANETs in such a way as to improve proper operation of the networks. A trust model for directory based service discovery is proposed using the evidential theory, Dezert-Smarandache Theory (DSmT). The DSmT [5] is an extension of the classical Dempster-Shafer Theory (DST) [4,6,7] for plausible and paradoxical information. The DST cannot

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 115–126, 2014.

handle paradoxical information. We address both the uncertain and paradoxical behavioral information during the fusion of the evidence collected from multiple nodes. The trust value is computed using the DSmT, based on the behavior of the mobile nodes. We perform this analysis of packets on our service discovery approach, DBF-SSD (Dynamic Bloom Filter based Semantic Service Discovery) [8]. In the DBF-SSD, the service discovery approach is based on the directory-based architecture, where these directories (i.e., cluster-head nodes) are selected, based on our clustering protocol, the AETCP [9]. During the service discovery process, the trust values of the service requester nodes are computed, directly or indirectly, by the service provider nodes.

The rest of this paper is organized as follows. Section 2 discusses the related work. The application of the Dezert-Smarandache Theory in trust evaluation for directory-based service discovery for MANETs is discussed in Section 3. The simulation results are shown in Section 4. Finally, Section 5 concludes this paper with future work.

2 Related Work

Various trust based approaches for mobile ad hoc networks have been proposed in the literature and analyzed in recent survey papers [1, 10, 11]. Since, this paper is mainly based on trust computations between the service provider and service requester nodes, we restrict our literature to works that deal with the similar issues.

The Secure Pervasive Discovery Protocol (SPDP) [12] is a fully distributed protocol, which provides the location of trusted services and secures communication between nodes. A trust relationship between nodes is established, based on the node's personal opinions (direct trust) and based on recommendations from trusted third parties (indirect trust). The service discovery middleware is proposed [13], to deliver stable services based on the trust of devices in the network. It includes service and delivery managers and trust level managers. A service provision framework is proposed for pervasive computing [14] to assess the QoS and the reliability of peers to choose the best service providers.

A Trust-based Secure Service Discovery (TSSD) is proposed for a truly pervasive environment [15]. The TSSD system includes the device discovery unit, the Service discovery agent and the trust, risk, and security unit. This system calculates the direct and indirect trust values of the service requester nodes. Głowacka et al. proposed a security system, based on the Dezert-Smarandache theory (DSmT) to enhance security in tactical MANETs [16]. This system handles uncertain, incomplete and conflicting information on the behavior of nodes from different sources. This system is presented with illustrations with no further simulation analysis.

Compared to the security addressed by Głowacka et al. [16], our proposed model uses the DSmT, to obtain legitimate decisions about the service requester nodes, as a result of fusion of paradoxical information from different sources of evidence. Our model investigates the direct trust values of service requester nodes in the presence of on-off attacks. The adaptive time factor is used in the DSmT to improve the packet delivery ratio in the presence of on-off attacks.

3 Proposed Trust Model

3.1 Overview of Our Approach

Assume that all nodes are mobile, and clusters are formed, based on the AETCP [9]. Then the service discovery approach, the SSD-DBF [8] is applied, to allow the exchange and sharing of each other's services, where the cluster-head nodes act as service directory (SD) nodes. Each SD node maintains a registry to store the service information, and a Dynamic Bloom Filter (DBF) [17], as an index, to summarize its contents based on the service operation names [8]. Hence, in the proposed trust system, each node may be a service provider node, a service requester node, a service directory node or an intermediate node.

We embed the trust mechanism, between the service requester and the service provider nodes, based on the DSmT [5,18,19], into our SSD-DBF approach. The trust values are computed based on the information that one node can collect about the other, by analyzing the received and forwarded packets. The information concerning the successful transmission of any packets can be obtained by any node using the passive acknowledgements method. In this method, after the transmission of any packet, the sender node analyzes the behavior of the next hop by placing itself in the promiscuous mode [20]. Often, each node observes its immediate neighbors, and records the number of positive and negative events. The positive events correspond to the generation of successful service replies, timely forwarding of packets, generation of successful acknowledgments, etc. Conversely, the negative events correspond to the abnormal forwarding of service requests and replies, refusal of packet forwarding to save energy, abnormal generation of service requests and replies, etc.

The service requester formulates and sends a query to the service directory node in its cluster. According to the SSD-DBF approach, the requested services are discovered locally or globally. When any node requests for a service, the trust computations are performed in the following ways.

1. Trust evaluation is performed on the requested node by the corresponding cluster-head node (i.e. service directory node) in the following ways:

   - If the service directory node S_1 never interacted with the requested node A (w.r.t. service related interactions) and has no other information of A, then S_1 should represent the trust for A as *(0,0,0,1)* (here, *(0,0,0)* represents the degree of trust, distrust, and contradiction respectively; *1* represents the degree of uncertainty).

   - If the service directory node S_1 interacts with the requested node A, then S_1 should represent the trust for A as direct trust.

   - If the service directory node S_1 never interacted with the requested node A, and has information of A from other nodes that connect A, then S_1 should represent the trust for A as recommendation trust.

2. Similarly, the trust evaluation is performed on the corresponding cluster-head node (i.e., service directory node) of the requested node by the service provider node in

the same manner, as explained in 1. The trust evaluation mechanism is explained in the next subsection.

3. If the service requester node is a trusted one, then the service information is sent to the requester node directly from the corresponding SD node.

3.2 Trust Evaluation Mechanism Based on DSmT

In ad hoc networks, the trust between nodes is evaluated, based on the service – based interactions. The trust evaluation is elucidated in the following steps.

Step 1: Representation of trust

Let $\Theta = \{T, \neg T\}$ be the frame of discernment, where T and $\neg T$ represent trust and distrust respectively. The hyper-power set D^{Θ} is defined as the set of all composite possibilities built from Θ with $\cup$ (conjunction) and $\cap$ (disjunction) operators. The hyper-power set D^{Θ} is defined as $D^{\Theta} = \{\phi, \{T\}, \{\neg T\}, \{T \cap \neg T\}), \{T \cup \neg T\}\}$. The theory of evidence, the DSmT, assigns a belief mass to each element of the hyper-power set. A function, $m : D^{\Theta} \to [0,1]$ is called a general basic belief assignment (*gbba*).

The *gbba* of the empty set is zero,

$$m(\phi) = 0 \tag{1}$$

The *gbba* of the remaining elements of the hyper-power set is totalled to 1, which supports paradoxical information.

$$\sum_{A \in D^{\Theta}} m(A) = 1 \tag{2}$$

That is, $$m(\{T\}) + m(\{\neg T\}) + m(\{T \cap \neg T\}) + m(\{T \cup \neg T\}) = 1 \tag{3}$$

where $m(\{T\})$ is the degree of trust, $m(\{\neg T\})$ is the degree of distrust, $m(\{T \cap \neg T\})$ is the degree of contradiction, and $m(\{T \cup \neg T\})$ is the degree of uncertainty. Thus, we employ, the trust for any mobile nodes in ad hoc networks represented as a 4-tuple, $(\{T\}, \{\neg T\}, \{T \cap \neg T\}), \{T \cup \neg T\})$, where each elements in the tuple ranges from 0 to 1.

Step 2: Trust Computation

Step 2.1: Direct Trust Computation

The trust is calculated between any two nodes that are directly connected and interact between them. For instance, the trust computation is performed between any service requester node and the corresponding service directory node, and between the service provider node and the service directory node of the requesting node.

The mobile node *M*'s opinion about the mobile node *N*'s behavior (direct trust evaluation) [21] is evaluated to obtain the degree of trust, distrust and uncertainty, respectively, as

$$m_N^M(\{T\}) = \frac{p}{p+n+2} \tag{4}$$

$$m_N^M(\{\neg T\}) = \frac{n}{p+n+2} \tag{5}$$

$$m_N^M(\{T \cup \neg T\}) = \frac{2}{p+n+2} \tag{6}$$

where p and n represent the number of positive and negative events, respectively, about mobile node N countered by mobile node M.

Due to the dynamic nature and characteristics of MANETs, the degree of contradiction is also considered, as an intrinsic measure, where the inconsistency may occur between $m_N^M(\{T\})$ and $m_N^M(\{\neg T\})$ values. The degree of contradiction of node N is calculated by node M, as follows:

$$m_N^M(\{T \cap \neg T\}) = -[m_N^M(\{T\}) \times \log_2(P) + m_N^M(\{\neg T\}) \times \log_2(Q)] \tag{7}$$

where $$P = m_N^M(\{T\}) + m_N^M(\{T \cup \neg T\}) \tag{8}$$

and $$Q = m_N^M(\{\neg T\}) + m_N^M(\{T \cup \neg T\}) \tag{9}$$

After normalizing these sets of values (4-tuple opinions), the sum of the 4-tuple opinions is set to 1.

$$m_N^M(\{T\}) + m_N^M(\{\neg T\}) + m_N^M(\{T \cap \neg T\}) + m_N^M(\{T \cup \neg T\}) = 1 \tag{10}$$

The node that evaluates the direct trust, maintains both the new and previous (history) set of trust values, in its record. Thus, the current direct trust CDT_N^M value is calculated using equations, and is represented as $\left(m_N^M(\{T_c\}), m_N^M(\{\neg T_c\}), m_N^M(\{T_c \cap \neg T_c\}), m_N^M(\{T_c \cup \neg T_c\})\right)$. Also, the direct trust value is recalculated according to the history records. Thus, the direct trust DT_N^M is updated, using the history record as follows.

$$DT_N^M = \beta \times HDT_N^M + (1-\beta) \times CDT_N^M \tag{11}$$

where HDT_N^M is the direct trust value in the history records, and is represented as $\left(m_N^M(\{T_h\}), m_N^M(\{\neg T_h\}), m_N^M(\{T_h \cap \neg T_h\}), m_N^M(\{T_h \cup \neg T_h\})\right)$. DT_N^M is represented as $\left(m_N^M(\{T\}), m_N^M(\{\neg T\}), m_N^M(\{T \cap \neg T\}), m_N^M(\{T \cup \neg T\})\right)$. β is the adaptive time factor [22], used to assign weights to history and current information.

$$\beta = \begin{cases} \beta_s, m_N^M(\{T_h\}) \geq m_N^M(\{T_c\}) \\ \beta_l, m_N^M(\{T_h\}) < m_N^M(\{T_c\}) \end{cases}, \quad 0 < \beta_s < \beta_l < 1 \tag{12}$$

where $m_N^M(\{T_h\})$ and $m_N^M(\{T_c\})$ are the trust elements of HDT_N^M and CDT_N^M, respectively. If the trust value stored in the history records is lesser than the current one, then more importance is given to the value stored in the history record (using β_l) and vice versa (using β_s).

Step 2.2: Recommendation Trust Computation

We employ recommendation as the simplest form of trust propagation, where the computed trust on any of the target nodes gets propagated in the network. For instance, if node *M* gets to know the trust value of node *N* through nodes *X* and *Y*, then node *M* can actually avoid the explicit trust computation on node *N*, due to the lack of infrastructure, mobility, and autonomy in a MANET environment. The recommendation is based on the transitive property of trust.

Mathematically, if node *M* trusts node *X*, represented as a 4-tuple, $T_X^M = \left(m_X^M(\{T\}), m_X^M(\{\neg T\}), m_X^M(\{T \cap \neg T\}), m_X^M(\{T \cup \neg T\})\right)$ and node *X* trusts node *N*, represented as a 4-tuple, $T_N^X = \left(m_N^X(\{T\}), m_N^X(\{\neg T\}), m_N^X(\{T \cap \neg T\}), m_N^X(\{T \cup \neg T\})\right)$, then node *M* trusts node *N*, through trust transitivity, represented as 4-tuple, $T_N^M = \left(m_N^M(\{T\}), m_N^M(\{\neg T\}), m_N^M(\{T \cap \neg T\}), m_N^M(\{T \cup \neg T\})\right)$, which is defined as

$$T_N^M = T_X^M \otimes T_N^X \tag{13}$$

where
$$m_N^M(\{T\}) = \left(m_X^M(\{T\}) + \alpha m_X^M(\{T \cap \neg T\})\right) \times m_N^X(\{T\}) \tag{14}$$

$$m_N^M(\{\neg T\}) = \left(m_X^M(\{T\}) + \alpha m_X^M(\{T \cap \neg T\})\right) \times m_N^X(\{\neg T\}) \tag{15}$$

$$m_N^M(\{T \cap \neg T\}) = \left(m_X^M(\{T\}) + \alpha m_X^M(\{T \cap \neg T\})\right) \times m_N^X(\{T \cap \neg T\}) \tag{16}$$

$$m_N^M(\{T \cup \neg T\}) = 1 - \left(m_N^M(\{T\}) + m_N^M(\{\neg T\}) + m_N^M(\{T \cap \neg T\})\right) \tag{17}$$

α is the discount factor [19]. When $\alpha = 0$, node *M* keeps the most conservative trust to *X*'s referral.

Step 3: Trust Aggregation

When the trust value of a particular target node is propagated through multiple paths, multiple versions are received at the destination. Now, the aggregation operation at the destination can combine these values together to obtain a single trust value. Trust aggregation is essential for the final outcomes. The aggregation operation must follow the three notions: (1) the aggregation operation should be commutative and associative (2) the outcome of the aggregation must maintain the same tendency and support it, when the evaluations of the trust values from two different nodes have the same tendency and (3) the degree of contradiction in the outcome should increase, when the evaluations of the trust values from two different nodes are very different. For instance, the aggregation of the two paradoxical trust evaluations from two different nodes generates an evaluation of contradiction.

We use the Dezert-Smarandache's rule of Combination for trust aggregation, that follows the above three notions. For instance, the combination rule from two sets of trust values from nodes *X* and *Y* about the target node *T*'s behavior, is calculated as follows.

$$m_{X,Y}(A) = (m_X \oplus m_Y)(A) = \sum_{B,C \in D^{\Theta}, B \cap C = A \neq \phi} m_X(B) m_Y(C) \tag{18}$$

Since D^{θ} is closed under $\cup$ and $\cap$ operators, this rule of combination can always be used for the fusion of paradoxical or rational trust values from two different nodes.

Step 4: Decision making on trust

We carried out decision making on trust between the service provider and the service requester nodes, say, *M* and *N* respectively. Node *M* will trust or distrust node *N*, and tend to send the requested service information in the following three cases.

1. Assume that node *M* has interacted with node *N* (Direct trust evaluation). If $m_N^M(\{T\}) > b_t$ (where b_t is the belief threshold), then node *M* trusts node *N*. Otherwise, node *M* distrusts node *N*.
2. Assume that node *M* has not interacted with node *N*, but node *M* has some recommendations of node *N*. Through recommendations and DSmT's combination rule, if $m_N^M(\{T\}) > b_t$, then node *M* trusts node *N*. Otherwise, node *M* distrusts node *N*.
3. Assume that node *M* has no information of node *B*. So, $m_N^M(\{T \cup \neg T\}) = 1$, and node *M* is uncertain and trusts node *N*. In the beginning, these nodes do not know each other, and they have insufficient information to take the decision to interact. In the very beginning, thus, node *M* has a tendency to interact with node *N*. But, after several interactions, node *M* obtains more information of node *N*, and can evaluate the trust of node *N*.

3.3 Illustrations of the Proposed Trust Model Using the DSmT

Assume a network of nodes, in which nodes *M*, *D* and *N* are service provider, service directory, and service requester, respectively, as shown in figure 1. Suppose node *N* requests for a service to its service directory node *D*. If the service is available in the network, then the trust is evaluated for node N by its service directory node *D*. Since, the service directory node *D* does not have direct interaction with node *N*, it evaluates the trust through recommendation from other nodes, *X* and *Y*. The nodes *X* and *Y* are intermediate nodes between nodes *D* and *N*, respectively. Due to page limitations, the steps of the trust evaluation of these nodes using the DSmT, are not discussed here.

Thus, the final set of values of node N evaluated by node M, is given as,$m_N^M(\{T\}) = 0.681$, $m_N^M(\{\neg T\}) = 0.022$, $m_N^Y(\{T \cap \neg T\}) = 0.123$ and $m_N^M(\{T \cup \neg T\}) = 0.174$. Assume that the belief threshold, $b_t = 0.5$. Now, the node *M* takes a decision on node *N*. Among these sets of values, the degree of trust, $m_N^M(\{T\}) = 0.681$, is strengthened. Thus, the service directory node *M* trusts the service requester node *N*.

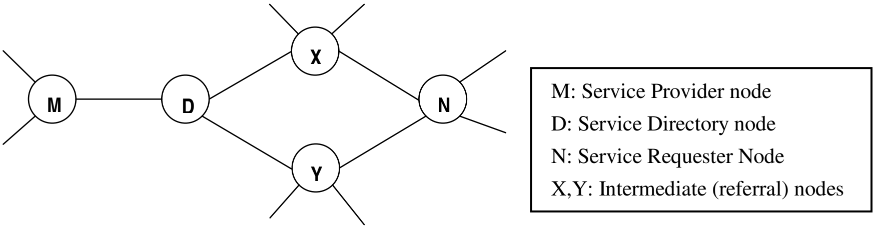

X
M
D
N
Y
M: Service Provider node
D: Service Directory node
N: Service Requester Node
X,Y: Intermediate (referral) nodes

Fig. 1. Nodes participating in trust evaluation

4 Simulation Results

The simulations were carried out using ns2 [23] to compare the performance of our trust system with that of the existing system. The performance metrics were, a transmission range of 300 m with a minimum speed of 1 m/s and maximum speed of 15 m/s, and a pause time of 20 s, using the modified version of the Random waypoint mobility model. The current random waypoint model fails to attain a steady state, in terms of the instantaneous average speed, and the speed continuously decreases, and hence, cannot be directly used for simulation. The improved model of the random waypoint mobility for the mobile nodes in the network is adopted, and the positive minimum speed is specified, in addition to, the maximum speed value [24,8,9].

The trust components of the proposed system, namely, the trust, distrust, contradiction and uncertainty, are measured and analyzed with/without the presence of attacks. Then, the packet delivery ratio is also analyzed, in a similar manner. The proposed system is compared with the existing approach SPDP [12], in terms of the service discovery success ratio and the packet delivery ratio.

4.1 Effects and Defense of On-off Attacks

The proposed trust model is applied to the civil unrest scenario [8]. In this scenario, the communication among the firefighter nodes, police personnel nodes and paramedic nodes is established in an ad hoc manner, to provide various functionalities, such as the rescue operations, detection/monitoring hazard activities, detecting the source of fires, retrofitting operations, riot control activities, etc. These nodes can be the service provider nodes, the service requester nodes, the service directory nodes or the intermediate nodes. Sometimes, a few service requester nodes, act as malicious nodes in an ad hoc network, and may indirectly aggravate the riot, depending on the importance of the situation, with indirect support from civilians / outsiders (not belonging to the ad hoc network of nodes). Thus, these malicious nodes may alternatively behave well and ill, and remain undetected, as they disrupt services, leading to an on-off attack. To handle this attack, the opinion made a long time ago about the nodes, should not include the same weight as that of the recent one, in order to avoid the selection of nodes that behave badly. Hence, this type of attack can be successfully handled by using the adaptive time factor (β) in the proposed trust computing system.

If the trust value stored in the history records is lesser than the current one, then more importance is given to the value stored in the history record, using the adaptive time factor, $\beta_l = 0.8$. Similarly, if the trust value stored in the history records is more than the current one, then less importance is given to the value stored in the history record, using $\beta_s = 0.3$. In the following simulation results, a few of the service requester nodes, acts as malicious nodes, and are considered for trust evaluation.

Figure 2 shows the components of the direct trust value of the service requesters with no attackers in the ad hoc networks. The associations between these components with the variation of time can also be discussed, from this figure. The trust component $m(\{T\})$ increases slowly, as the distrust $m(\{\neg T\})$ relatively decreases. Meanwhile, $m(\{T \cap \neg T\})$ represents the contradiction between the components $m(\{T\})$ and $m(\{\neg T\})$. Using the DSmT, the contradiction parameter, avoids giving a high trust value to the service requester nodes; instead, the direct trust value of the service requester nodes increases gradually (assuming the behavior of nodes to be good), as shown in Figure 2. When the trust component is relatively low, the contradiction $m(\{T \cap \neg T\})$ increases. At the end of the simulation time of 1000 s, the trust components of the service requester nodes are increased with the variation of time, the contradiction $m(\{T \cap \neg T\})$ decreases and reaches approximately zero.

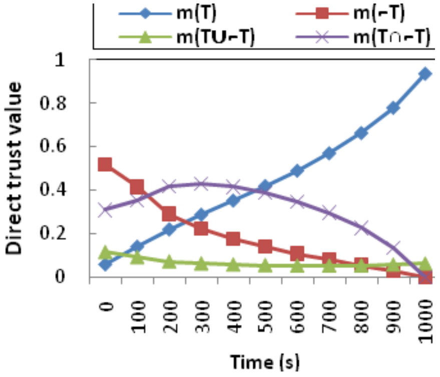


Fig. 2. Relationship between trust components without attackers

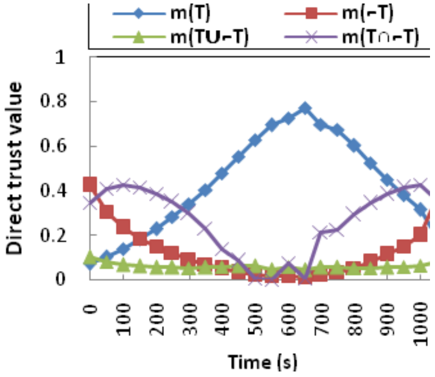


Fig. 3. Direct trust values in the presence of on-off attacks

Figure 3 shows the direct trust values of the malicious service requester nodes with the variation of time, in the presence of on-off attacks. Assume the malicious nodes' behavior in the following two stages: (1) first, they behave well till the end of a simulation time of 650 s, (2) then, they behave badly (from time 651s) till the end of simulation time of 1000 s. The components, $m(\{T\})$ increase slowly and $m(\{\neg T\})$ decreases slowly, as the malicious nodes behave well till the end of the simulation time of 650 s. Here, more importance is given to the value stored in the history record, using the adaptive time factor $\beta_l = 0.8$. Once the malicious service requester nodes behave badly, $m(\{T\})$ decreases sharply and $m(\{\neg T\})$ increases sharply, till the end of the simulation time. Here, less importance is given to the value stored in the history record, using $\beta_s = 0.3$. Thus, the proposed trust model defends against on-off attacks effectively, as the trust values increase slowly in the trust compensation period, and decrease sharply, once the malicious nodes behave badly. Also, using the adaptive time factor, the bad behaviors of the malicious nodes are remembered for a longer time, compared to the good behaviors. Still, the malicious nodes can recover their trust value after a number of bad behaviors, but this recovery needs more number of good actions.

Figure 4 shows the packet delivery ratio (PDR) with the variation of time in the presence of on-off attacks. The packet delivery ratio is considered as the percentage of packets that are successfully transmitted. Figure 4 shows three cases: (1) The packet delivery ratio is effectively improved with no attackers and no trust management. (2) The packet delivery ratio with no trust system, where attackers drop packets that pass through them. In this case, the malicious attackers significantly decrease the PDR. (3) The PDR is effectively improved using the proposed trust system, in the presence of on-off attacks. In this case, using the adaptive time factor and DSmT, the PDR is effectively improved and approximates the case (1).

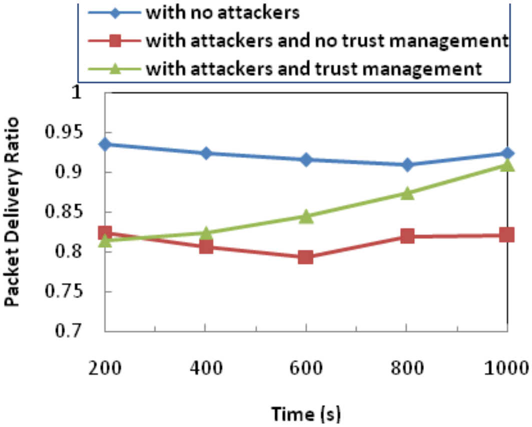


Fig. 4. Packet delivery ratio with trust management

Figure 5 shows the service discovery success ratio of the SPDP and the DSmT-based Trust, by varying the number of nodes from 60 to 100 for the transmission range of 300 m, with the minimum speed as 1 m/s and the maximum speed as 13 m/s. During the simulation period, the mobile nodes reach a steady-state average speed. Irrespective of the presence of attackers, the service discovery success ratio of the DSmT-based trust model is relatively improved compared to that of the SPDP. As shown in Figure 6, the packet delivery ratio of the DSmT-based trust model is also improved, compared to that of the SPDP. The DSmT-based Trust model uses the adaptive time factor to identify the attackers. The SPDP finds it hard to identify the attackers; accordingly, the discovery success ratio and the packet delivery ratio are affected.

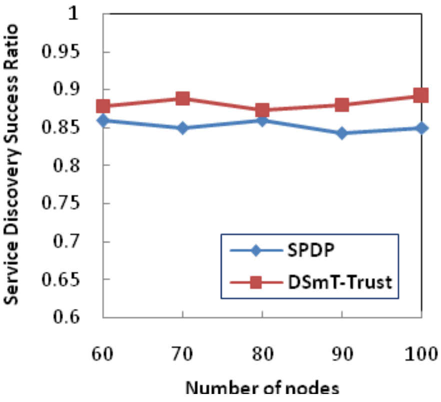


Fig. 5. Service discovery success ratio

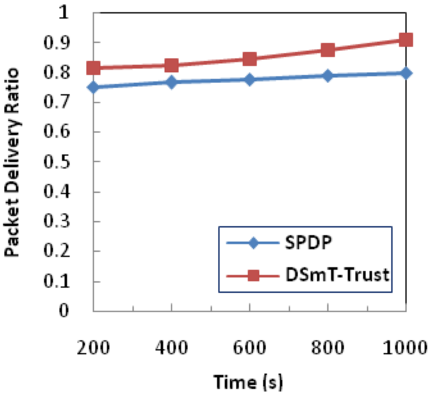


Fig. 6. Packet delivery ratio

5 Conclusion

This paper presents a framework for the trust evaluation of ad hoc networks, using an evidential theory, the Dezert-Smarandache Theory. Since mobile nodes are vulnerable to threats, the trusts between nodes are established for service discovery. In the proposed system, the DSmT is used to evaluate the direct and indirect trust computations of the service requester nodes. The mobile nodes may provide different opinions about the target service requester nodes, leading to paradoxical information. Using the DSmT rule of combination, legitimate decisions about these nodes can be taken, from the fusion of this paradoxical information. Simulations are performed to investigate the direct trust values of service requester nodes in the presence of on-off attacks. The adaptive time factor is used in the DSmT, to improve the packet delivery ratio in the presence of on-off attacks. Compared to the existing approach, the service discovery success ratio and the packet delivery ratio of the DSmT-based trust model are relatively improved, in the presence of attackers. In future work, notion of false recommendations with other types of attacks may be investigated.

References

1. Cho, J.H., Swami, A.: A Survey on Trust Management for Mobile Ad Hoc Networks. IEEE Commun. Surv. Tut. 13(4), 562–583 (2011)
2. Cho, J.H., Swami, A., Chen, I.R.: Modeling and Analysis of Trust Management for Cognitive Mission-Driven Group Communication Systems in Mobile Ad Hoc Networks. In: 12th IEEE International Conference on Computational Science and Engineering, pp. 641–650. IEEE Press, New York (2009)
3. Cook, K.S. (ed.): Trust in Society. Russell Sage Foundation Series on Trust, New York (2003)
4. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, New Jersey (1976)

5. Dezert, J.: Foundations for a New Theory of Plausible and Paradoxical Reasoning. Information and Security 9, 13–57 (2002)
6. Dempster, A.P.: Upper and Lower Probabilities Induced by Multivalued Mapping. Annals of Mathematical Statistics 28, 325–339 (1967)
7. Sentz, K., Ferson, S.: Combination of Evidence in Dempster-Shafer Theory. Sandia National Laboratories, Albuquerque, New Mexico (2002)
8. Deepa, R., Swamynathan, S.: The DBF-based Semantic Service Discovery for Mobile Ad Hoc Networks. Canadian Journal Electrical and Computer Engineering (accepted for Publication, 2014)
9. Deepa, R., Swamynathan, S.: AHP-Entropy-TOPSIS based Clustering Protocol for Mobile Ad Hoc Networks. Ad Hoc & Sensor Wireless Networks. Old City Publishing (accepted for publication, 2014)
10. Govindan, K., Mohapatra, P.: Trust Computations and Trust Dynamics in Mobile Ad Hoc Networks: A Survey. IEEE Commun. Surv. Tut. 14(2), 279–298 (2012)
11. Kumar, A., Aggarwal, A.: Lightweight Cryptographic Primitives for Mobile Ad Hoc Networks. In: Thampi, S.M., Zomaya, A.Y., Strufe, T., Alcaraz Calero, J.M., Thomas, T. (eds.) SNDS 2012. CCIS, vol. 335, pp. 240–251. Springer, Heidelberg (2012)
12. Campo, C., Almenárez, F., Díaz, D., García-Rubio, C., López, A.M.: Secure Service Discovery based on Trust Management for Ad-hoc Networks. J. Univer. Comput. Sci. 12(3), 340–356 (2006)
13. Han, S., Kim, J.M., Bin, H.: Service Discovery and Delivery System Based on Trust in Mobile Ad-Hoc Network. In: International Conference on Information Science and Security, pp. 171–176 (2008)
14. McNamara, L., Mascolo, C., Capra, L.: Trust and Mobility Aware Service Provision for Pervasive Computing. In: 1st International Workshop on Requirements and Solutions for Pervasive Software Infrastructures (co-located with Pervasive 2006), Dublin, Ireland, pp. 603–610 (2006)
15. Ahamed, S.I., Sharmin, M.: A trust-based secure service discovery (TSSD) model for pervasive computing. Computer Communications 31(18), 4281–4293 (2008)
16. Glowacka, J., Amanowicz, M.: Application of Dezert-Smarandache theory for tactical MANET security enhancement. In: IEEE Military Communications and Information Systems Conference, pp. 1–6 (2012)
17. Guo, D., Wu, J., Chen, H., Yuan, Y., Luo, X.: The Dynamic Bloom Filters. IEEE T. Knowl. Data En. 22(1), 120–133 (2010)
18. Smarandache, F., Dezert, J.: Advances and Applications of DSmT for Information Fusion, Collected works. Arp Publishers (2004)
19. Wang, J., Sun, H.J.: A new evidential trust model for open communities. Comput. Stand. Inter. 31(5), 994–1001 (2009)
20. Pirzada, A.A., McDonald, C.: Establishing trust in pure ad-hoc networks. In: 27th Australian Conference on Computer Science, pp. 47–54 (2004)
21. Jøsang, A., Ismail, R.: The Beta Reputation System. In: 15th Bled Conference on Elec-tronic Commerce, pp. 17–19 (2002)
22. Feng, R., Che, S., Wang, X., Yu, N.: Trust Management Scheme Based on D-S Evidence Theory for Wireless Sensor Networks. Int. J. Distrib. Sens. N, 1–9 (2013)
23. The Network Simulator, ns-2, available from World Wide Web, `http://nsnam.isi.edu/nsnam`
24. Yoon, J., Liu, M., Noble, B.: Random Waypoint Considered Harmful. In: 22nd Annual Joint Conference on IEEE Computer and Communications, pp. 1312–1321 (2003)

Chain Routing for Convergecast Small Scale Wireless Sensor Networks

C.R. Yamuna Devi[1], Deepak Sunder[1], S.H. Manjula[1],
K.R. Venugopal[1], and Lalit M. Patnaik[2]

[1] Department of Computer Science and Engineering
University Visvesvaraya College of Engineering,
Bangalore, India
[2] Indian Institute of Science, Bangalore, India
yamuna_devicr@yahoo.com

Abstract. Wireless sensor networks have many applications involving autonomous sensors transmitting their data to a sink placed in the network. A protocol by name Chain Routing for Convergecast Small Scale (CRCSS) Wireless sensor networks is proposed in this paper. The set of sensor nodes in the network send the data periodically to the sink located in the area of interest. The nodes who cannot reach sink in one hop choose one of the neighbours for forwarding the data to the sink by forming a chain of links. The selection of forwarding node and the waiting period before forwarding plays an important role in the protocol. The proposed CRCSS protocol exhibits improvement in energy spent per packet and latency per packet for a wireless sensor network as compared to ConverSS protocol for small scale wireless sensor networks. In CRCSS protocol energy spent per packet is independent of the network radius.

Keywords: Communication System, Convergecast Routing, Energy, Latency, Multi-hop Networks**.**

1 Introduction

Wireless sensor network is a collection of sensor nodes deployed to monitor an area of interest in the environment. The nodes in the sensor network sense a parameter of interest in the network and report it to the base station or the sink located in the network. The nodes can be homogeneous or heterogeneous in terms of transmission range, initial energy stored, mobility, etc. The nodes are connected to the sink through a single hop or more than one hop through intermediate nodes. The routes formed prior to the start of packet communication from the sensor node to the sink are called static routes. Dynamic routes between the source and the sink are the routes that are formed as and when an intermediate route is reached. Two types of links between the sensor nodes are possible. In symmetric links the transmission power and receive threshold are same in both direction of the data flow between any two sensor nodes. Transmission power and receive threshold vary in case of asymmetric nodes.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 127–138, 2014.

Wireless sensor networks find applications in many fields of human life. Initially wireless sensor networks were used in the Military fields for surveillance application. Now wireless sensor networks find applications in field such as intruder detection in any restricted area, weather monitoring, health monitoring, etc. Automobile industry is another area where wireless sensor networks are used for tracking of vehicles and as an information system. It is possible for a bus passenger to know the location of the bus he is waiting for and to know whether he can get a seat in the bus or not when he gets into the bus in his stop. The network size and network area depends on the type of application of the wireless sensor network.

The sensor nodes comprise of a sensing unit, a memory unit, a transmitting unit, a receiving unit and a central processing unit. The sensing unit is responsible for sensing the parameter of interest and transfers it to processing unit. Memory unit present in sensor nodes is smaller in size as the data sensed is not bulk in size and there is no need to store the sensed data in the memory for longer time. The sensed data is transferred to the processing unit or the sink. The transmitter is used for transmitting the sensed or the received data to another node or the sink. The function of receiving unit is receiving the data from a neighbouring node or receiving acknowledgement from the sink node. The processing unit does the minimal processing required on the data before it is transferred to the sink node.

The data sensed by the sensors is communicated to the sink in the specific location within the network which is known as convergecast routing. In convergercast routing the route leading to sink will be congested as all the sensor nodes have to communicate their data to the single sink. Route congestion results in loss of data and longer latency period in packet transmission. Fig. 1 shows an example of convergecast network. Node 0 is the sink node. The remaining nodes send the data if they have to the sink in their sending slots. Nodes 1, 2, 3, 4and 5 are one hop distant from the sink node. Nodes 6, 7, 8, 10 and 11 are two hops away from the sink node. Packet from node 12 takes three hops to reach the sink node. If all the nodes have data to send to the sink in all the sending slots congestion occurs around node 0. This congestion cannot be completely eliminated but it can be reduced.

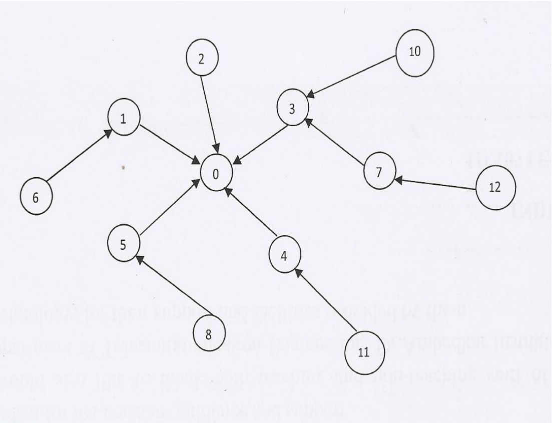


Fig. 1. Convergecast Routing

The CRCSS is a combined MAC and routing solution for reliable and energy-efficient convergecast for small-scale sensor networks. CRCSS is designed specifically for cases where most nodes are one to three hops away from the sink. The protocol proposed in this paper Chain Routing for Convergecast Small Scale wireless sensor networks has two fold contributions. First it reduces the energy spent by the sensor nodes in the network irrespective of the number of nodes in the network. Second it decreases the latency per packet transmission from source node to the sink node.

The rest of the paper is organised as follows. Section 2 lists and discusses the papers related to convergecast routing in wireless and other networks. Background work needed for proposing the new protocol is discussed in Section 3. Problem definition and algorithm are explained in Section 4. Section 5 analyzes the results of network simulation. Finally the paper concludes in section 6.

2 Literature Survey

This section discusses some of the papers related to convergecast routing, their advantages and disadvantages.

Chen et al., [1] present a novel convergecast tree protocol and a distributed algorithm to attain load balancing among the nodes of the network and to extend network lifetime. Dynamic adjustment of convergecast tree structure avoids breaking tree link, and is controlled by sensor's grandparent to avoid looping problems. The adjustment mechanism is localized and does not require global information. The simulations performed on convergecast networks demonstrate that throughput is increased. Fu et al., [2] design two cooperative schemes called convergimo schemes, for static and mobile ad hoc wireless networks. Static convergimo, utilizes Multiple Input Multiple Output (MIMO) technology to turn interfering signals into interference-resistant ones. Mobile convergimo characterizes on joint transmission from multiple nodes to multiple receivers, to maximize the throughput.

Hong and Kim [3] propose the Express-MAC, EX-MAC, protocol to conserve energy in a wireless sensor network using asynchronous duty cycling and to decrease end-to-end latency in packet transmission using wakeup time reservation. The EX-MAC protocol supports multi-hop network applications through a cross-layering interface, and provides convergecast packets with unidirectional interfaces to optimize performance and to support reconfiguration routing. Bernson and Manivannan [4] discuss design the factors of Vehicular Ad Hoc Network routing protocols. The authors classify and characterize the existing greedy routing protocols and provide a qualitative comparison of all the routing protocols with respect to their objectives, design approaches and requirements. The approaches discussed focus on dense traffic scenarios in wireless sensor networks.

Kam and Schurgers [5] propose a combined MAC and routing protocol ConverSS that uses contention-free MAC in conjunction with beaconing and overhearing in small scale convergecast wireless sensor networks. The protocol is optimized to handle single-hop networks, but, has the ability to route a packet through multiple hops, when necessary. An improvement of a factor of 10 is observed by ConverSS routing protocol, as compared to ideal protocol for small scale wireless sensor networks. X. Zhang et al., [6] analyze the performance bounds of typical many-to-one communication represented by convergecast scheduling, oriented to industrial applications. Three scenarios are considered in analyzing the networks. They are the lower bound on number of time slots to finish intra-cluster and inter-cluster convergecast transmissions, lower bound on the number of channels based on the number of timeslots and available channels and packet re-transmissions to meet the reliability requirements.

H. Zhang et al., [7] study time-optimal convergecast under the communication constraints of commodity sensor network platform. The authors propose a novel convergecast model in which packet copying between the processor and transceiver are separated from packet transmission. Both centralized and distributed schemes are proposed for computing time-optimal convergecast schedule. Augustine et al., [8] study the network in which the sensor nodes send information to the sink node as a byte packet. The path between the sensor node and the sink is the shortest one and the sensor node packs the data sensed into fully packed packets before sending it to the sink. This protocol reduces the energy consumed by nodes in packet transmission as the packet count is reduced.

Theoleyre [9] propose C-MAC protocol for multi-hop convergecast wireless network that selects a sub-tree of the shortest paths to the sink containing exactly k leaves to forward maximum of the traffic towards the sink. C-MAC is based on CSMA-CA like approaches and assigns priorities to the k-tree core nodes to avoid collisions among themselves to increase the throughput compared to original IEEE 802.11 protocols. Advantage of this protocol is that it does not require any synchronization mechanism among the sensor nodes in the network.

3 Background

Convergecast for Small Scale (ConverSS) networks [5] is a combined MAC and routing solution for small-scale, mobile networks. ConverSS is designed specifically for cases where maximum of the nodes are one hop away from the sink. For many of these real-time sensing applications, sensor generated data must be sent to the sink periodically. In one cycle of the protocol operation, or a sending interval, each node has one data packet to deliver to the sink. Because these are small networks, it is feasible to use a fixed-assignment TDMA MAC, in which each node is assigned a dedicated time slot for sending. This MAC requires that the number of nodes in the network be fixed at the system initialization. However, this setup is sufficient for most sensing missions, which have a small, consistent team of vehicles.

ConverSS [5] operates under two phases in each sending slot. Every node is given a time slot to send its packet to the sink based on Time Division Multiple Access Technique. Phase 1 operates under the assumption that nodes are one hop from the sink. Each node attempts to send the packet directly to the sink in its sending slot. Since there is no routing in this phase, nodes do not need to listen in the other nodes slots in phase I. They can instead be placed in sleep mode, in which nodes turn off the radio and thus consume very low power. All nodes check the Delivery Status Bit (DSB) sent by the sink to discover if their data have been delivered. In case of packet errors, more frames are allotted in which nodes can retransmit any undelivered data. By the end of phase 1, packets from all 1-hop nodes should have been received by the sink. If this includes all nodes, then they all go to sleep until the next sending interval. The operation described is efficient because there is no route setup, and nodes only send in their own slots and do not need to listen in the other slots.

It is possible that there are still undelivered packets after Phase 1 because of packet errors or nodes being out of range of the sink. In Phase 2, any nodes whose packets were not delivered to the sink perform a type of controlled flooding, in which nodes broadcast their data and receiving nodes can rebroadcast to try to deliver it to the sink. The reason for using flooding rather than a route setup followed by data transmission relates to the presence of asymmetric links. Fig. 2 shows the header details for ConverSS protocol. Studies have demonstrated the problem of asymmetry in radio propagation, in which a link is stronger in one direction than in the reverse. In typical routing, only symmetric links can be used because a handshake is required before a packet can be sent over a link.

The ConverSS header consists of the following fields: Packet Type, Packet Sequence Number, Source Address, Destination Address, Delivery Status Bit (DSB) and Acknowledgment Bitmap (ACKB). The type of packet is a data packet or acknowledgement packet. The Delivery Status Bit indicates the source node’s knowledge of which nodes’ data have been successfully delivered. The Acknowledgment Bitmap indicates from which nodes it has received since its last sent packet. Every data packet uses this header.

Packet Type	Seq. No.	Source Address	Destination Address	DSB	ACKB

Fig. 2. ConverSS Header Fields

Flooding is used during the phase 2 of packet transmission from sensor nodes to the sink. Flooding does not require a handshake, thus enabling the use of asymmetric links in routing the data to the sink. Since these are small networks, routes with asymmetric links may be the only ones available. Therefore, with more options for routing, data delivery has an improved chance of success. After the two phases are completed, the network can go to sleep until the next sending interval. As the network has small number of nodes, the two-phase sending interval is typically short enough so that the network topology will be stable during that time. Because no routes are assumed prior to Phase 1, the protocol is robust to changes in topology in between sending intervals. The sensor nodes cannot go to sleep until all the nodes have sent the data to the sink. A sensor node might be required to forward the packet to the sink. This results in higher consumption of energy by the sensor nodes. The proposed CRCSS constructs chain of links from source node to the sink in the network ensuring the delivery of packet to the sink. The proposed protocol reduces this waiting period by the sensor nodes by using the location of the sensor nodes to find the number of hops required to reach the sink. This reduction in waiting period reduces the energy consumption by the nodes and reduces the latency of packet transmission.

4 Problem Definition

In the proposed CRCSS protocol the two phases of ConverSS [5] are combined into a single phase to reduce the waiting period of the sensor nodes in the wireless sensor networks. The objective of Chain Routing for Convergecast Small Scale wireless sensor networks is to reduce the energy consumed by the network and latency per packet as compared to ConverSS routing.

The following points are assumed in the simulation of the wireless sensor network under consideration.

1) The sensor nodes in the network are homogeneous nodes.
2) The sink has infinite energy and transmission distance.
3) Small size network topology is considered up to 50 sensor nodes.
4) The transmission range of sensor nodes is fixed to 100 meters.
5) The network is error free.

Table 1. Algorithm CRCSS: Chain Routing for Convergecast Small Scale Wireless Sensor Networks

```
Input: Wireless sensor network with sink and sensor node positions.
Output: Chain of links from each of  sensor node to the sink.

wake up at the start of sending slot
for all the sending slots do
  if current time!= end of sending slot then
    if I have data packet to be sent then
      if $D_{N,S} <= T_r$
        then
          send packet to $S$
          Delivery Status Bit DSB = 1
          $P = S$
        else
          find the neighbor whose DSB=1
          if such node exists then
            route the packet through that node
          endif
          if (DSB=1) for more than one node
            choose the node nearest to sink.
          endif
          $P = N$
       endif
     else
       sleep until the next sending interval
    endif
  endfor
```

Table 1 explains the algorithm CRCSS to find the chain of shortest path from sensor nodes in the network to the sink. Wireless sensor network with sink S and other sensor node *N* positions is the input to the algorithm. In each sending slot Ts the sensor node sends the data to the sink node. The source node goes to sleep till the next sensing slot, if it does not have any data to send to the sink. The sensor node finds the distance between itself and the sink node $D_{N,S}$. If this distance is within Ts then the packet can reach the sink in one hop and the sink becomes the parent of the sensor node.

On the other hand, if the sink is not reachable from the source, the source node selects among its neighbours the ones that have sink as the parent node. One of them is allowed to forward the packet to the sink and the forwarding node becomes the parent of the sending node. This process continues till all the sensor nodes in the network send their data packet to the sink and all the sending slots are completed. After the completion of the algorithm the path from sensor nodes to sink is defined. Energy consumed by the nodes, average number of hops in transmission and latency of packet delivery are analyzed.

Figure 3 shows an example convergecast wireless sensor network of 12 nodes. Node 0 is the sink node and the remaining nodes send the data to the sink periodically. Number of hops varies from one hop to three hops depending on the distance from the sensor node to the sink and the transmission range of the sensor nodes. The sensor nodes are initially supplied with an initial energy. The sensor nodes consume energy for sensing, receiving and transmitting packets. Small amount of energy is spent by sensor nodes in idle mode and sleep mode. The nodes will spend energy for all their activities over time.

5 Performance Analysis

The network is simulated using Network Simulator NS2 to analyze the behaviour of sensor nodes. Three different network scenarios are considered to analyze the performance of the proposed Chain Routing for Convergecast Small Scale wireless sensor networks protocol. In the first case network is of 11 sensor nodes and a sink is considered. Node 0 is the sink node and the other nodes send the data to the sink at regular intervals. Second network considers sensor nodes numbering from 5 to 40 to observe change in the latency per packet.

Finally, in the last case wireless sensor network radius is varied from 100m to 160m to measure the saving in energy spent per packet transmission by the sensor nodes during the simulation time. The sensor nodes in all the three cases are initially loaded with an uniform energy of 5Joules. All the wireless sensor nodes are deployed in the area of 500m by 500m. Energy is spent as and when the nodes transmit and receive packets during simulation in the network. The duration of the simulation time is taken as 100 seconds for all the cases.

5.1 Case 1

The protocol CRCSS is implemented on a 12 node wireless sensor network shown in Fig. 3 to analyze the energy spent per packet transmission between the source nodes and sink. Transmission energy and reception energy are measured per packet transmission. Sleep energy is measured per unit time such as energy spent by a sensor node in sleep mode per second. Table 2 shows the energy model parameters values used in the simulation environment.

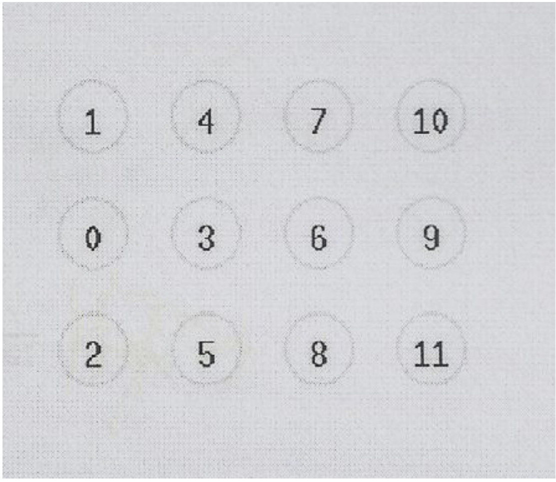


Fig. 3. Convergecast network of 12 nodes

Table 2. Energy Model Parameter Values

Parameter	Symbol	Value
Initial Energy	E_I	5J
Transmission Energy	E_T	0.14J
Reception Energy	E_R	0.095J
Idle Energy	E_D	0.08J
Sleep Energy	E_S	0.06J

Figure 4 shows the energy spent by sensor nodes to transmit a packet to the sink node for different power values in case of ConverSS [5] and the proposed CRCSS protocols. Sensor power values 1 and 2 are representing different transmission ranges of the sensor nodes as 50m and 100m respectively. For power =1 or transmission distance of 50m, the reduction of energy spent by sensor nodes for CRCSS is less than that of existing ConverSS [5] by 9mJ when all the 11 sensing nodes are considered. A reduction of 14mJ is obtained when the number of sensing nodes is 11 for power =2 or transmission distance of 100m.

5.2 Case 2

A wireless sensor network with size $N =$ 12 nodes with base station in the centre is considered with number of nodes increased up to 40. The transmission range Tr of the wireless sensor nodes is fixed to 100m. Data packets are transmitted from the sensor nodes to the sink at the rate of one packet per second. Energy model for this network is as given in Table 2. The packet latency L and energy spent per packet transmission Ep are analyzed for this network setup. Latency is the time interval between the sending of the packet at the source node and the reception of the packet at the destination node. A sensor node spends energy in sensing, transmitting, receiving and forwarding operations. The energy spent in sensing an event is very less compared to transmission energy and reception energy and hence it is neglected in the energy computations.

Graph in Fig. 5 shows the latency per packet in case of convergecast(CON) protocol and CRCSS protocol. Up to 25 nodes the sensor nodes are connected to the sink through single hop. As the number of hops between source and sink nodes is same, latency per packet transmission is same for both CON and CRCSS protocols. For more than 25 nodes CRCSS reduces the number of hops as compared to CON routing. As a result of this reduction a decrease of 70% and 88% in latency per packet is observed when the number of nodes is 30 and 40 respectively in case of CRCSS protocol.

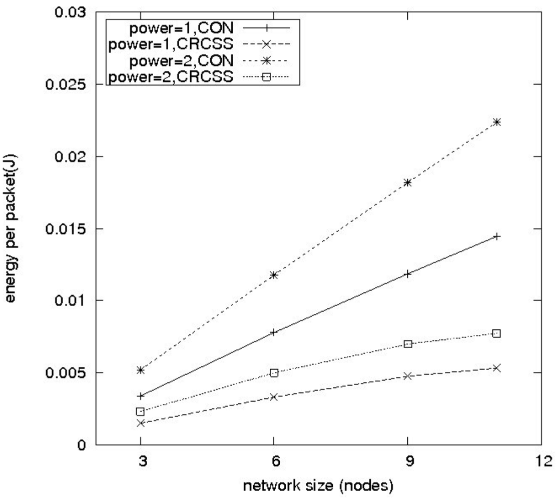


Fig. 4. Network Size versus energy spent per packet

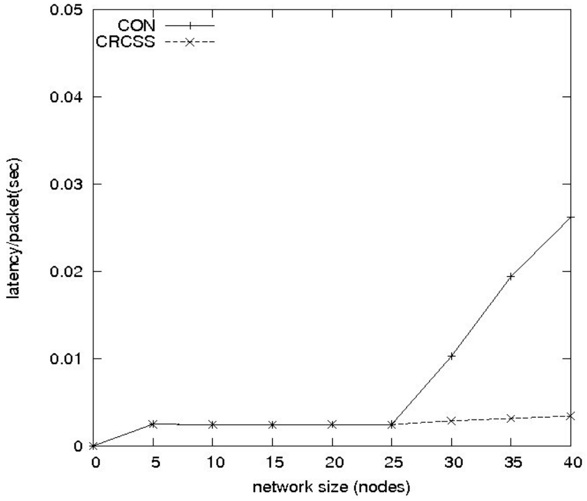


Fig. 5. Network Size versus Latency Per Packet

Figure 6 shows the energy spent per packet when data is sent from source node to sink for network size varied up to 40. The initial energy of the sensor nodes is 5J. The nodes consume energy for different operations in the network. The number of hops remains same in existing convergecast and the proposed CRCSS up to 25 nodes because major of the communication requires one hop. The same number of hops results in same energy spent per packet transmitted between source and sink. Transmission range of the nodes is 100 m and all the nodes are within one hop distance from the sink. For 30 and 40 nodes energy spent in CRCSS protocol is lesser by 20% and 36% respectively as compared to convergecast routing protocol. The reason for the reduction energy spent by the nodes in packet transmission is the decrease in the waiting time of the nodes in the case of CRCSS protocol.

5.3 Case 3

This subsection studies the effect of network radius *R* on the energy spent per packet transmission in a wireless sensor network. Keeping the number and density of sensor nodes as constant the network radius is varied to analyze the energy spent in the network. With the increase in the network radius the distance between the sensor nodes increases correspondingly. This results in reduction in number of hops between the source and the sink in the network. Fig. 7 shows the energy spent per packet when the network radius is increased from 100m to 160m. For less than 100m network radius all the nodes in the network can send their data to the sink in one hop that results in uniform energy consumption by all the sensor nodes in the network. A decrease of 4mJ of energy spent per packet transmission is seen from the graph when the network radius is 100m. For network radius of 160 m reduction in the energy spent for transmission of one packet to the sink is 18mJ.

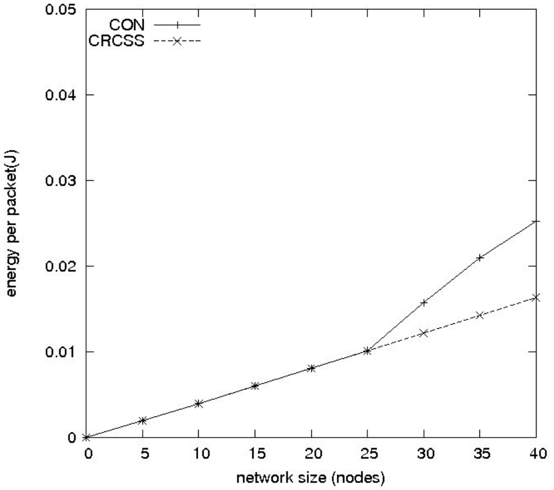


Fig. 6. Network Size versus energy spent per packet

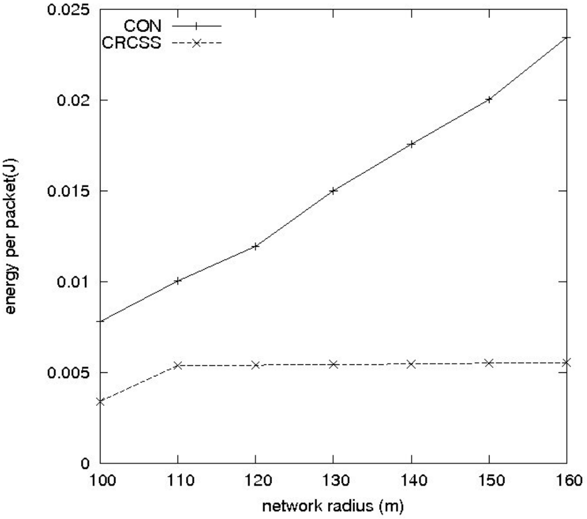


Fig. 7. Network Radius versus Energy Spent per Packet

6 Conclusion

A new protocol CRCSS is proposed for small scale wireless sensor networks in this paper. The proposed CRCSS protocol uses controlled flooding instead of broadcasting. Sending of packets from the source to sink is done by forming a chain from the source to the sink when they are more than one hop away. The proposed CRCSS protocol exhibits up to 36% improvement in energy spent per packet and up to 88% improvement in latency per packet for 40 nodes network. It is proved by simulation that the network radius does not affect energy spent per packet in CRCSS protocol. Analysis of the energy, latency and number of hops in the routing for mobile sensor nodes is considered for future work. Another direction for future work is to consider network with errors and attackers in the network. Parameters like current energy of the node and path length to the source can be considered in the selection of the forwarding node.

References

1. Chen, T.-S., Tsai, H.-W., Chu, C.-P.: Adjustable Convergecast Tree Protocol for Wireless Sensor Networks. Elsevier Journal on Computer Communications (33), 559–570 (2010)
2. Fu, L., Qin, Y., Wang, X., Liu, X.: Throughput and Delay Analysis for Convergecast with MIMO in Wireless Networks. In: Proceedings of IEEE INFOCOM, pp. 1–20 (2011)
3. Hong, S.-H., Kim, H.-K.: A Multi-hop Reservation Method for End-to-End Latency Performance Improvement in Asynchronous MAC-based Wireless Sensor Networks. IEEE Transactions on Consumer Electronics 55(3), 1214–1220 (2009)
4. Bernsen, J., Manivannan, D.: Greedy Routing Protocols for Vehicular Ad Hoc Networks. In: Proceedings of International Wireless Communication and Mobile Computing Conference, IWCMC 2008 (2008)
5. Kam, C., Schurgers, C.: ConverSS: A Hybrid MAC/Routing Solution for Small-Scale Wireless Networks. IEEE Transactions on Mobile Computing 10(9), 1227–1236 (2011)
6. Zhang, X., Liang, W., Yu, H., Feng, X.: Optimal Convergecast Scheduling Limits for Clustered Industrial Wireless Sensor Networks. International Journal of Distributed Sensor Networks, 1–12 (October 2012)
7. Zhang, H., Osterlind, F., Soldati, P., Voigt, T., Johansson, M.: Rapid Convergecast on Commodity Hardware: Performance Limits and Optimal Policies. In: 7th Annual IEEE Secon, Sensor Mesh and Ad Hoc Communications and Networks (SECON), pp. 1–9 (2010)
8. Augustine, J., Han, Q., Loden, P., Lodha, S., Roy, S.: Tight Analysis of Shortest Path Convergecast in Wireless Sensor Networks. International Journal of Foundations of Computer Science 24(1), 31–50 (2013)
9. Theoleyre, F.: A Route-Aware MAC for Wireless Multihop Networks with a Convergecast Traffic Pattern. The International Journal of Computer and Telecommunications Networking 55(3), 822–837 (2011)

A Genetic Algorithm for Scheduling Workflow Applications in Unreliable Cloud Environment

Lovejit Singh and Sarbjeet Singh

Computer Science and Engineering, UIET, Panjab University,
Chandigarh, India
pu.lovejitjhajj@gmail.com, sarbjeet@pu.ac.in

Abstract. Cloud Computing refers to application and services offered over Internet using pay-as-you-go model. The services are offered from data centers all over the world, which jointly are referred to as the "Cloud". The data centers use scheduling techniques to effectively allocate virtual machines to cloud applications. The cloud applications in area such as business enterprises, bio-informatics and astronomy need workflow processing in which tasks are executed based on data dependencies. The cloud users impose QoS constraints while executing their workflow applications on cloud. The QoS parameters are defined in SLA (Service Level Agreement) document which is signed between cloud user and cloud provider. In this paper, a genetic algorithm has been proposed that schedules workflow applications in unreliable cloud environment and meet user defined QoS constraints. A budget constrained time minimization genetic algorithm has been proposed which reduces the failure rate and makespan of workflow applications. It allocates those resources to workflow application which are reliable and cost of execution is under user budget. The performance of genetic algorithm has been compared with max-min and min-min scheduling algorithms in unreliable cloud environment.

1 Introduction

Cloud Computing offers computational and storage resources on demand by using pay-as-you-go model. The computational and storage resources are provided with the help of virtualization technologies. The cloud applications in areas such as business enterprises, bio-informatics and astronomy need workflow processing in which tasks are executed based on data dependencies. Cloud users generally impose QoS constraints while executing their applications on the cloud. These QoS parameters are defined in SLA (Service Level Agreement), which is signed between cloud user and cloud provider. The workflow applications may contain sensitive data that cannot tolerate failure of resource on which the applications are scheduled along with QoS constraints. With this motivation, we propose genetic algorithm which schedules workflow applications in unreliable cloud environment and meets user defined QoS constraints. The proposed genetic algorithm finds the schedule that costs the user under his budget and also reduces the failure rate and makespan of workflow

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 139–150, 2014.

application. The performance of the algorithm has been compared with list heuristic scheduling algorithms viz. max-min and min-min, in unreliable cloud environment.

The rest of the paper is structured as follow: The related work is presented in section 2. The problem overview is presented in section 3. The proposed approach is presented in section 4. Experimental results and comparison are presented in section 5 and Section 6 concludes the work carried out.

2 Related Work

Many heuristic and meta-heuristic approaches have been proposed by different researchers to schedule workflow applications in cloud. In heuristic approach, priority concept is used to schedule workflow applications. The list scheduling algorithms viz. max-min and min-min algorithms are based on heuristic approach which divides the scheduling process into two phases: task prioritizing and resource selection phase. In task prioritizing phase, a rank value is assigned to each task based on task priorities. In resource selection phase, a high priority task is selected and scheduled on optimal resources which complete the task earliest [1]. The meta-heuristic approach includes scheduling algorithms which are based on iteration method to find the solution to optimization problem. The genetic algorithm is meta-heuristic scheduling algorithm which uses evolution process to find the best schedule for workflow application. It finds the solution that optimizes the execution of entire workflow application whereas heuristic algorithms find the schedule for workflow application based on task level. Scheduling algorithms mostly focus on optimization of cost and makespan. The genetic algorithms presented in [2]-[8] reduce makespan of workflow applications on cloud resources. The algorithms proposed in [9]-[11] optimize makespan and reduce failure rate of workflow application. In [12], a budget constraint genetic algorithm has been implemented which finds the schedule for workflow application under user budget as defined in SLA. We feel that there is a need to implement workflow scheduling algorithm which considers the reliability of cloud resources while allocating them to applications. With this motivation, work on budget constraint time minimization genetic algorithm has been carried out which provides reliable and cost effective machines to users for executing workflow applications under user budget.

3 Problem Overview

3.1 Problem Description

The workflow application is represented as a Directed Acyclic Graph (DAG) such as G (T, E) where, T is a set representing finite number of tasks and E is a set representing data dependencies among tasks. The tasks in workflow application can be executed level by level. The tasks on same level can be executed in parallel. Figure 1 shows a three level workflow graph.

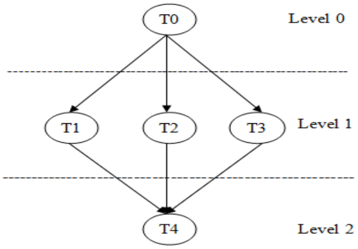


Fig. 1. Workflow Graph G

A cloud data center consists of number of physical machines. These physical machines are abstracted with the help of virtualization technologies and represented as virtual machines. Each virtual machine charges a different cost to user. A virtual machine having high MIPS rate and high reliability charges more to user than the virtual machine having low MIPS rate and low reliability. Scheduling algorithms are expected to find a map for every task in set T with virtual machines such that users QoS constraints are met. They must return the schedule that costs the user under his budget and reduces failure rate and makespan of workflow application.

3.2 Genetic Algorithm Approach

Genetic algorithms are inspired by the evolution process of nature. A genetic algorithm combines solutions from past searches and explore new range of solutions. Each solution in genetic algorithm is represented in terms of an individual. In initial population generation stage, numbers of individuals are generated randomly and each individual is kept different from other on the basis of fitness value. A better individual has high fitness value than other individuals. Typically, a genetic algorithm consists of following steps.

A Creation of individuals in initial population. (This is generally done randomly. The size of initial population depends on the nature of problem.)

B Generation of new individual by applying selection. (In this, parents are selected and crossover and mutation operations are applied to generate new off springs.)

C Evaluation of the fitness value of newly generated individual.

D Repetition of the step 2 and 3 until stopping criteria is met and then best individual in returned.

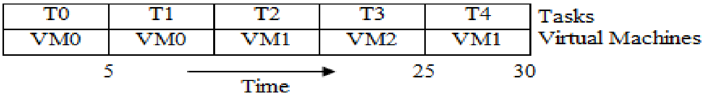
Following sections describe how genetic approach has been applied to workflow scheduling problem.

4 Proposed Approach

4.1 Genetic Algorithm

The proposed genetic algorithm consists of following steps:

A *Encoding:* It has been done using identification number of each virtual machine and the task. In Figure 1, there are five tasks in a workflow application. Each task has a unique identification number from 0 to 4. Assume there are 3 virtual machines each having unique identification number from 0 to 2. Figure 2 shows encoding individuals in the population by using identification number of tasks and virtual machines.



T0	T1	T2	T3	T4	Tasks
VM0	VM0	VM1	VM2	VM1	Virtual Machines

Fig. 2. Individual

B *Initial Population Generation:* In initial population, individuals are generated randomly. The reliability of randomly selected virtual machine is checked before scheduling tasks on it. The task t is selected from set T that has not been allocated to any virtual machine. Then a virtual machine (vm) is selected randomly form set VM. If selected vm is reliable then schedule the task t on vm otherwise select another vm randomly from set VM. This process continues until individual is generated. The numbers of individuals are generated in initial population according to the size of initial population. The quality of individuals obtained in initial population depends upon threshold. The threshold value is set according to workflow application requirement. If the workflow application could not be afforded to fail then the value of reliability threshold should be high. This selects only those virtual machines for scheduling which are highly reliable.

C *Evaluation:* In evaluation step, the fitness value is calculated on the basis of makespan, cost and reliability. An individual having minimum makespan, minimum cost and high reliability will have high fitness value compared to other individuals. So following fitness function has been used in the current implementation:

Fitness function (F) = R (I) / (M (I) * C (I))

In the above formulae, M (I) is make span of individual, C (I) is cost of individual and R (I) is reliability of individual.

Makespan of individual M (I) $= \sum_{i=0}^{n} T_{et_i}$

Task execution time (T_{et}) = (Instruction length of task) / (MIPS rate of virtual machine)

Cost of individual C (I) $= \sum_{i=0}^{n} C_{ec_i}$

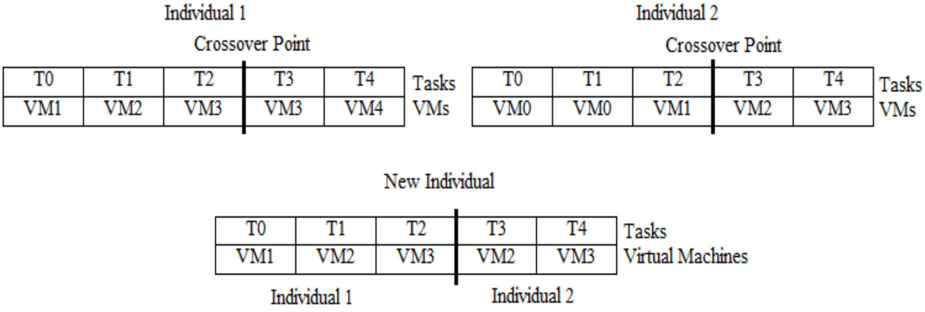
Task execution cost (C_{ec}) = (Instruction length of task) * cost scaling factor

Cost scaling factor is a constant which depends on the number of instructions in the task. The cost scaling factor is more for a task having less number of instructions and less for a task having more number of instructions.

Reliability of individual R (I) = min (R (VM_{t1}), R (VM_{t2}) , … R (VM_{tn})) where R(VM_{t1}) is the reliability of vm where first task is to be executed, R(VM_{t2}) is the reliability of vm where second task is to be executed, and so on. $R(VM_{tn})$ is the reliability of vm where n^{th} task is to be executed.

D *Selection:* In selection process, two schedules (called parents) are selected from the population through a selection method. In the proposed algorithm, roulette wheel selection method has been used because there are more chances to select that individual who has highest fitness value. The roulette wheel contains slots whose size is relative to fitness value of individual. That means individual with higher fitness will have bigger slot size as compare to individual with lower fitness. Linear search is performed when individuals are selected from roulette wheel. The wheel is twisted 2N times, where N is twice the number of individuals in the population. Crossover of two individual will result in a new individual. Therefore we need to twist wheel twice to pick two individuals randomly for crossover. On each twist, the individual under the wheel's marker is selected to act as parent for next generation.

E *Crossover:* In crossover process, genes of two individuals are interchanged to produce new individual. The proposed algorithm uses single point crossover where two parent individuals transfer their genes at corresponding point and produce new individual by selecting the first half from first parent and second half from second parent as shown in Figure 3.



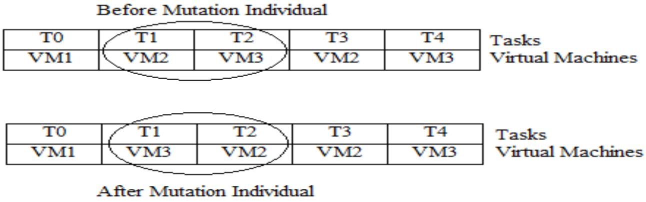
T0	T1	T2	T3	T4	Tasks
VM1	VM2	VM3	VM3	VM4	VMs

T0	T1	T2	T3	T4	Tasks
VM0	VM0	VM1	VM2	VM3	VMs

T0	T1	T2	T3	T4	Tasks
VM1	VM2	VM3	VM2	VM3	Virtual Machines

Fig. 3. Single Point Crossover

F *Mutation:* In mutation process, random change is performed in child schedule. Two randomly picked tasks interchange their virtual machines in child schedule. Figure 4 shows child schedule after applying mutation process on schedule in Figure 3.



T0	T1	T2	T3	T4	Tasks
VM1	VM2	VM3	VM2	VM3	Virtual Machines

T0	T1	T2	T3	T4	Tasks
VM1	VM3	VM2	VM2	VM3	Virtual Machines

Fig. 4. Mutation

In Figure 4, swap operator was used to interchange the virtual machines position in child schedule. The mutation operator explores new region of solution while performing random swapping in child schedule.

G *Termination:* The algorithm terminates when number of generations reach user given threshold. The number of iteration to find best schedule depends upon threshold value. The genetic algorithm returns better quality individual (schedule) if the threshold value is set higher.

Figure 5 shows the flow of proposed genetic algorithm. The pseudo code of genetic algorithm is given below:

```
1   Initialize workflow application and virtual machines.
2   While (No. of individuals < initial population size)
3       Set the status of all tasks in set T to unscheduled to generate new individual
4       While (Set T has unscheduled tasks)
5           Select a task t from set T of tasks which has not been allocated to any
            virtual machine. The tasks are picked according to their dependencies.
6           Select a virtual machine vm randomly from set VM of virtual machines.
7           If (reliability of vm > workflow app reliability) then
8               Schedule task t on virtual machine vm
9           Else
10              Go to step  6
11          End if
12      End while
13      Store the new generated individual in population P.
14  End while
15  Evaluate fitness of all individuals in population P.
16  While (No. of population generated < population threshold)
17      While (New population size < twice of original population size)
            Select parent using roulette wheel selection
19          Apply single point crossover
20          If (Mutation rate > mutation threshold) then
21              Apply mutation
22          End if
23          Add generated off-springs to current population
24      End while
25      Evaluate fitness of all individuals in new population.
26      Based on fitness value of individuals, keep topmost individuals in new
        population equal to original population size.
27  End while
28  Sort the individuals of final population in decreasing order according to their
    fitness value.
29  Select first individual from population i.e. the individual that has highest fitness
    value.
30  If (Selected individual's cost < workflow app budget) then
31      Return selected individual as best individual and exit
32  Else
```

```
33          Select next highest fitness value individual from population P if available
            and go to step 30.
34      End if
35      No feasible solution found as per user specified budget.
```

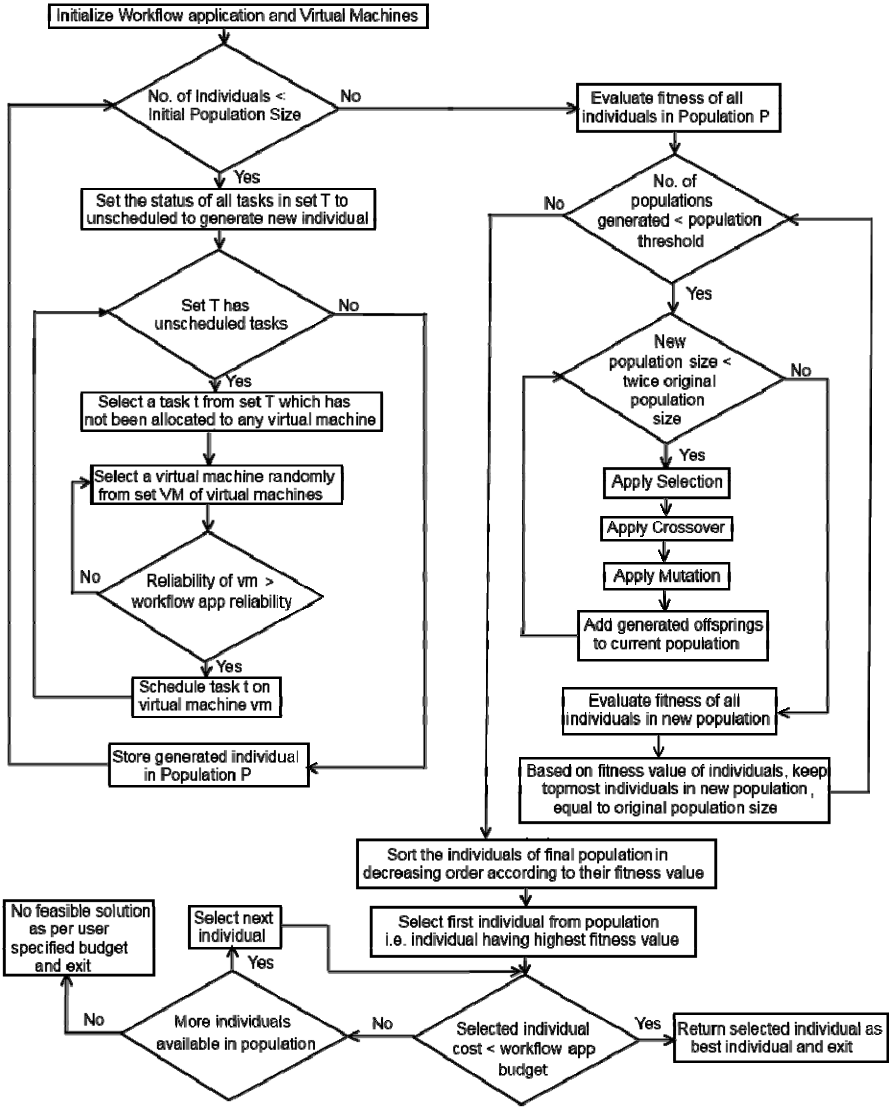

Initialize Workflow application and Virtual Machines
No. of Individuals < Initial Population Size
No
Yes
Set the status of all tasks in set T to unscheduled to generate new individual
Set T has unscheduled tasks
No
Yes
Select a task t from set T which has not been allocated to any virtual machine
Select a virtual machine randomly from set VM of virtual machines
No
Reliability of vm > workflow app reliability
Yes
Schedule task t on virtual machine vm
Store generated individual in Population P
Evaluate fitness of all individuals in Population P
No
No. of populations generated < population threshold
Yes
New population size < twice original population size
No
Yes
Apply Selection
Apply Crossover
Apply Mutation
Add generated offsprings to current population
Evaluate fitness of all individuals in new population
Based on fitness value of individuals, keep topmost individuals in new population, equal to original population size
Sort the individuals of final population in decreasing order according to their fitness value
Select first individual from population i.e. individual having highest fitness value
No feasible solution as per user specified budget and exit
Select next individual
Yes
No
More individuals available in population
No
Selected individual cost < workflow app budget
Yes
Return selected individual as best individual and exit

Fig. 5. Proposed Genetic Algorithm

5 Experimental Results and Comparison

We used CloudSim [13] to simulate cloud environment to perform our experiments. Table 1 provides information about simulator resources which were used to perform experiments i.e. number of data centers, hosts, virtual machine parameters etc.

Table 1. Simulation Parameters

Parameter	Value
Number of Datacenters	1
Number of Hosts	25
MIPS Rate of Host	10000
RAM of Host	2 GB
Number of VMs on each Host	2
RAM of VM	512 MB
Mode of VMs on Host	Time-Shared
Total number of VMs	50
Initial Population Generation	Randomly
Crossover Operator	Single-Point
Mutation Operator	Swapping
Initial Population Size	100
Number of Iterations	100
Crossover Probability	0.5
Mutation Probability	0.01

The experiment was conducted on 50 virtual machines in unreliable cloud environment which means certain virtual machines in the environment are highly unreliable. The virtual machine that was unreliable results into run time failure and the reliability of virtual machines are known to scheduler before execution. Figure 6 shows reliability distribution of virtual machines. The MIPS rate of each virtual machine is shown in Figure 7. There is diversity of virtual machines in the experiment. It varies from 800 to 4000 MIPS. The information about MIPS rate of virtual machines is known to scheduler before the execution. The scheduling algorithm estimates the execution time of tasks by using MIPS rate of virtual machines.

The experiment has been conducted on a workflow application which consists of 50 tasks. The workflow application has been executed using CloudSim and the performance has been compared with three scheduling algorithms viz. genetic algorithm, max-min algorithm and min-min algorithm. The performance evaluation has been done on the basis of three parameters: failure rate, cost and makespan.

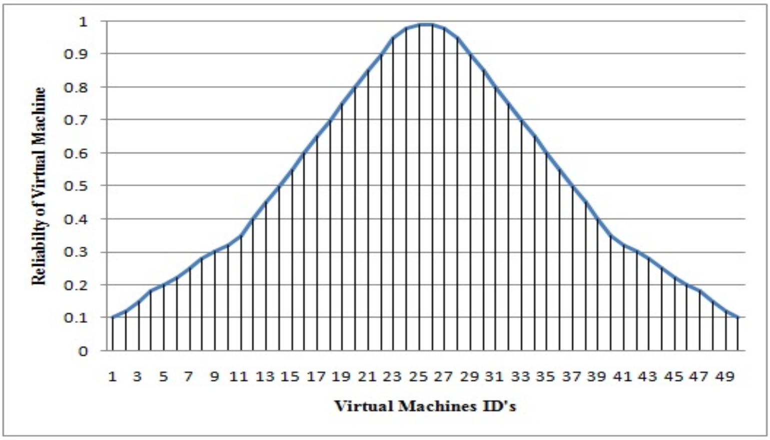


Fig. 6. Reliability Distribution

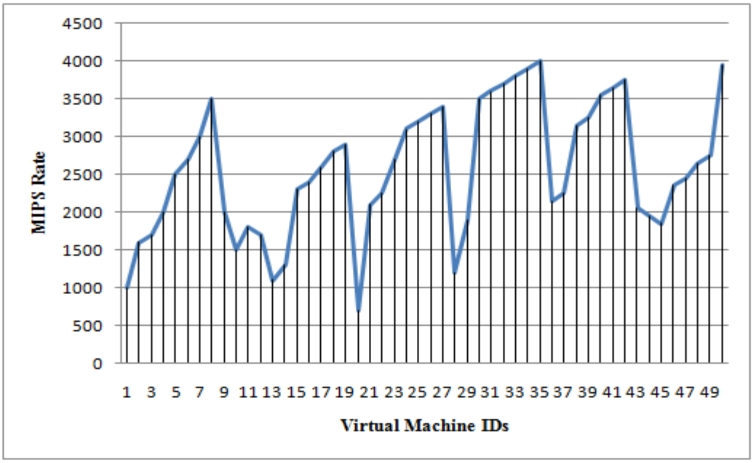


Fig. 7. Virtual Machines MIPS Distribution

A *Failure rate:* Reliability is the ability of virtual machine to perform and maintain its function in routine circumstances. The reliability of virtual machines can be measure by calculating the failure rate of workflow applications. In the current work, failure rate has been calculated by using the following formula:

Failure rate = F (I) / T (I)

Where, F (I) is the number of workflow applications that are failed on scheduled virtual machine. T (I) is total number of workflow applications that are scheduled on virtual machine.

The genetic algorithm allocates highly reliable virtual machines to workflow application tasks and reduces the failure rate of workflow application as compared to max-min and min-min algorithm as shown in Figure 8. It filter outs unreliable virtual machines during initial population generation phase.

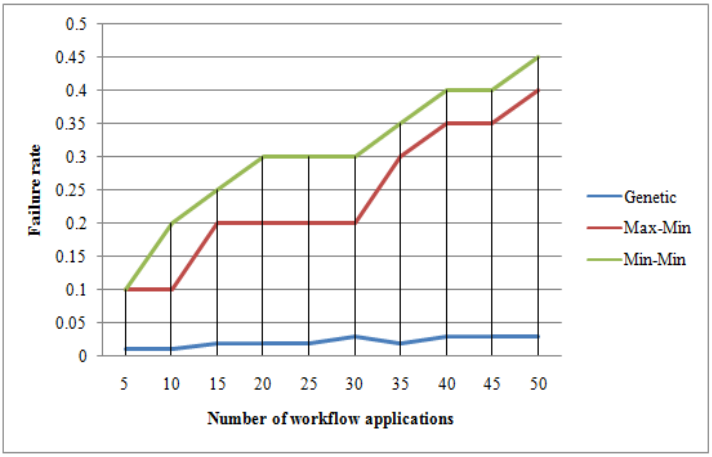


Fig. 8. Comparison of Failure Rate

B *Cost:* Genetic algorithm minimizes the execution time under user budget constraints. Table 2 shows budget of workflow application as defined by the user.

Table 2. User budget and execution cost of scheduling algorithms for workflow application

User Defined Workflow Application Budget	Rs. 1500
Genetic Algorithm Execution Cost	Rs. 1270
Max-Min Execution Cost	Rs. 1085
Min-Min Execution Cost	Rs. 838

In Figure 9, it is clear that max-min and min-min reduces execution cost of workflow application as compared to genetic algorithm.

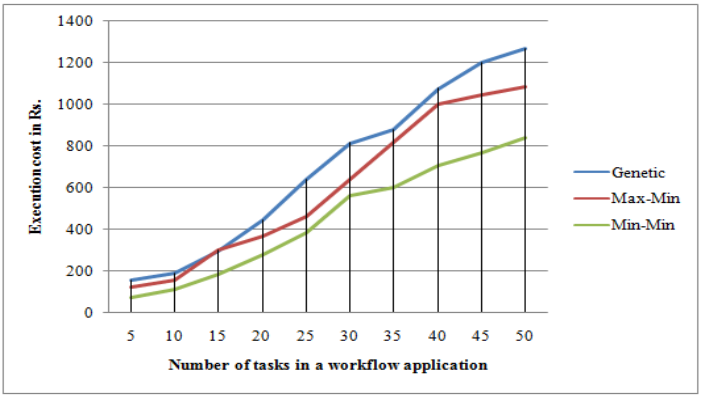


Fig. 9. Comparison of Execution Cost

C *Makespan:* The genetic algorithm reduces the makespan of workflow application as compared to max-min and min-min algorithms. It achieves better parallelism as shown in Figure 10. Genetic algorithm executes the workflow application in 818 seconds. It reduces the execution time as compared to max-min and min-min algorithms which need 1082 and 1370 seconds respectively.

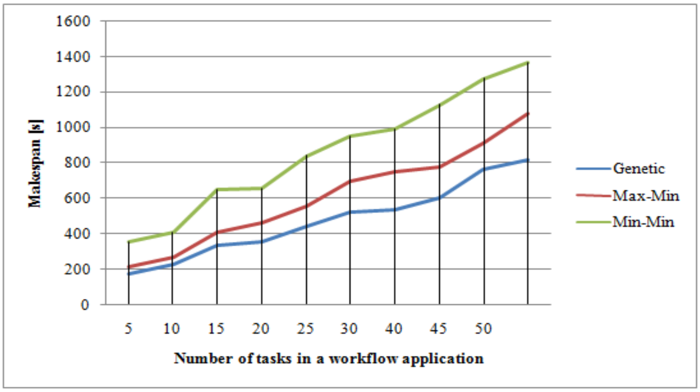


Fig. 10. Comparison of Makespan

6 Conclusions and Future Scope

In this paper the problem of scheduling workflow application in unreliable cloud environment has been handled. A budget constraint time minimization genetic algorithm for scheduling workflow applications in unreliable cloud environment has been simulated in CloudSim. The genetic algorithm considers virtual machines reliability along with user defined budget constraint while scheduling workflow application on virtual machines. It reduces failure rate and makespan of workflow applications. The performance of genetic algorithm has been compared with max-min and min-min scheduling algorithms in unreliable cloud environment. From the results obtained, it is clear that genetic algorithm finds the optimal schedule that meets the user budget constraint, reduce the execution time and failure rate of workflow application as compared to max-min and min-min scheduling algorithms. In future, we are planning to introduce load balancing concept which will increase the utilization of virtual machines and reduce the total number of virtual machines required to execute the workflow application.

References

1. Yu, J., Buyya, R., Kotagiri, A.: Workflow Scheduling Algorithms for Grid Computing, vol. 146, pp. 173–214. Springer, Heidelberg (2008)
2. Hou, E.S.H., Ansari, N., Ren, H.: A Genetic Algorithm for Multiprocessor Scheduling. In: IEEE Proceeding on Parallel and Distributed Systems, vol. 5 (1994)

3. Wang, P.C., Korfhage, W.: Process Scheduling using Genetic Algorithm. In: Parallel and Distributed Proceeding Seventh IEEE Symposium, pp. 638–641 (1995)
4. Wang, L., Siegel, H.J., Roychowdhury, V.P.: A Genetic Algorithm Based Approach for Task Matching and Scheduling in Heterogeneous Computing Environments. Journal of Parallel and Distributed Computing-Special Issue on Parallel Evolutionary Computing Archive 47, 8–22 (1997)
5. Liu, D., Li, Y., Yu, M.: A Genetic Algorithm for Task Scheduling in Network Computing Environment. In: Algorithms and Architectures for Parallel Processing Proceeding IEEE Fifth International Conference, pp. 126–129 (2002)
6. Page, A.J., Naughton, T.J.: Dynamic Task Scheduling using Genetic Algorithm for Heterogeneous Distributed Computing. In: Proceedings 19th IEEE Conference on Parallel and Distributed Processing Symposium (2005)
7. Moattar, E.Z., Rahmani, A.M., Derakhshi, M.R.F.: Job Scheduling in Multiprocessor Architecture using Genetic Algorithm. In: 4th IEEE Conference on Innovations in Information Technology, pp. 248–251 (2007)
8. Mocanu, E.M., Florea, M., Ionut, M.: Cloud Computing Task Scheduling Based on Genetic Algorithm. In: System IEEE Conference, pp. 1–6 (2012)
9. Dogan, A., Ozguner, F.: Bi-Objective Scheduling Algorithms for Execution Time and Reliability Trade off in Heterogeneous Computing System. The Computer Journal 48, 300–314 (2005)
10. Wang, X.F., Yeo, C.S., Buyya, R., Su, J.: Optimizing the Makespan and Reliability for Workflow Applications with Reputation and a Look-ahead Genetic Algorithm. Future Generation Computer Systems 27, 1124–1134 (2011)
11. Delavar, A.G., Aryan, Y.: A Goal-Oriented Workflow Scheduling in Heterogeneous Distributed System. IJCA 52, 27–33 (2012)
12. Yu, J., Buyya, R.: A Budget Constraint Scheduling of Workflow Application on Utility Grid Using Genetic Algorithm. In: 15th IEEE International Symposium on High Performance Distributed Computing (HPDC 2006), Paris (2006)
13. Calheiros, R.N., Ranjan, R., De Rose, C.A.F., Buyya, R.K.: CloudSim: A Novel Framework for Modelling and Simulation of Cloud Computing Infrastructures and Services. GRIDS Laboratory. The University of Melbourne, Australia (2009)

Autonomic SLA Management in Cloud Computing Services

S. Anithakumari and K. Chandra Sekaran

NITK, Manglore, Karnataka, India
{lekshmi03,kchnitk}@gmail.com

Abstract. Cloud computing has developed into a more acceptable computing paradigm for implementing scalable infrastructure resources given on-demand in a pay-by-use basis. Self-adaptable cloud resources are needed to meet users application needs defined by Service Level Agreements (SLAs) and to limit the amount of human interactions with the processing environment. Sufficient SLA monitoring techniques and timely discovery of possible SLA violations are of principal importance for both cloud providers and cloud customers. The identification of possible violations of SLA is done by analyzing predefined service level objectives together by using knowledgebases for managing and preventing such violations. In this paper we propose a new architecture for the detection of SLA violation and also for the re-negotiation of established SLAs in the case of multiple SLA violations. This re-negotiation of SLAs will really help to limit the over provisioning of resources and thus leads to the optimum usage of resources. As a consolidation the proposed architecture may yield maximized Business Level Objectives (BLOs) to the cloud providers.

1 Introduction

Cloud computing presents a new computing paradigm to implement scalable computing infrastructure by using concepts of virtualization of resources and distributed application design[2]. Cloud services are meant for high performance applications which really needs plenty of system resources. In cloud computing, service provisioning is based on Service Level Agreements (SLA), which is an agreement signed between the service provider and the customer. It clearly mention the terms of the service which includes the non-functional necessities of the service expressed concisely as quality of service (QoS), service pricing, obligations, and penalties for violation of agreements.

Management of SLA agreements, in a flexible and reliable manner, is equally important to both, cloud consumers and cloud providers. Prevention of SLA violations prior to its occurrence can eliminate unnecessary payment of penalties a provider has to give at the time of violations. In some cases, simple recovery actions such as VM migration may prevent SLA violations. By using flexible and timely reactions to possible violations of SLA, human interactions can be reduced to a maximum extent which increases the chance of cloud computing to prosper as a reliable and flexible on-demand computing paradigm.

To guarantee an already established SLA, the service provider should be able to continuously monitor the infrastructure resource metrics. Conventional monitoring mechanisms for individual or clusters of resources are limited to the area and homogeneity

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 151–159, 2014.

of examined items and so cannot be applicable in cloud in a suitable way. Also in conventional systems there is a visible difference between low-level entities like monitored metrics and high level parameters like SLA agreements[5]. So a major focus required for such SLA monitoring is, monitoring of resource metrics of cloud resources, and mapping of these resource metrics to SLA related parameters. Some major works[6] have tried in this area and as a result the researchers have come up with a general architecture for the management of SLAs related to cloud services. This architecture is called FoSII architecture which includes concepts about autonomic SLA management and implementation. The FoSII architecture contains a separate framework called LoM2HiS that is responsible for mapping the measured low level application metrics to the high level SLA parameters. By using the FoSII architecture, some SLA violation detection mechanisms are also devised which will detect the chance of occurrence of an SLA violation detection and provide mechanism, like provisioning of additional resources, to avoid such a violation. But none of these mechanisms tried for re-establishing the SLA upon the detection of several SLA violations.

In this paper we propose a novel architecture which focuses on the detection of the occurrence of SLA violation and also on the re-negotiation of established SLAs in the case of multiple SLA violations. This re-negotiation of SLAs will really help to reduce the over provisioning of resources and leads to the optimum usage of resources. The following sections describes the basic details about this newly proposed architecture and its mapping to the SLA negotiation strategy to go for re-negotiation of the established SLAs. The rest of the paper is organized as follows. Section 2 presents the related work in this field and Section 3 describes the details about the significance of SLA in cloud computing. Section 4 explains SLA and related metrics considered at the time of configuration of resources. Section 5 introduces our newly proposed architecture for autonomic SLA management. section 6 presents the details of the implementation planning. Finally section 7 concludes the paper and present the scope of future research works in this area.

2 Related Work

As of now only limited number of works has been done in the area of resource monitoring, low-level metrics mapping, and detection of SLA violations in cloud computing, and so the related areas like ervice-Oriented Architecture (SOA) and Grid related works are also studied to get a clear idea of SLA related processing. Wood et al.[12] introduced a new system, named Sandpiper, which automates the process of monitoring and identifying hotspots and remapping or reorganizing of VMs at the time of necessity. Boniface et al.[1] explain dynamic provisioning of services using SLAs. The authors also discuss provisioning of services according to agreed upon SLAs and the monitoring of SLAs for limiting violations. There they have considered only Grid environments and not computing clouds. Koller and Schubert [9] describe autonomous management of QoS parameters by using a proxy-like approach based on a WS-Agreement implementation. Thereby, SLAs can be exploited to design certain QoS parameters that a service has to maintain during its interaction with a particular customer. Foundations of Self-governing ICT Infrastructures (FoSII) is a developed research project in Vienna University of Technology [11]. It proposes concepts and models for autonomic management and enforcement of SLAs. Frutos and Kotsiopoulos[7] discuss the main approach of the EU project BREIN[3] to develop a framework that extends the characteristics of computational Grids by driving their usage inside new target areas in the business domain for advanced SLA management. BREIN applies SLA management to Grids, whereas we focus SLA management in clouds. Dobson and Sanchez-Macian[4] present a unified QoS ontology applicable to QoS-based Web services selection, QoS monitoring, and QoS adjustment. However they do not consider application provisioning and deployment strategies.

3 SLA and Cloud Computing

Cloud computing environments contain several cloud providers which provide similar cloud services/computing resources and so the consumer has to choose the most suitable provider for his needs. As of now the differentiating elements between cloud computing solutions are Quality-of-Service (QoS) and the Service Level Agreements (SLA) guarantee provided by the cloud providers. SLA denotes an agreement or contract signed by both the parties, the service provider and the customer, comprising non functional requirements of the service represented as QoS. QoS of the cloud include features such as performance (eg. service response time, throughput), availability of service and similar measures. SLA also includes obligations, details of service pricing and penalties for violations of agreements. Another important aspect considered with SLAs is the essential elasticity of Cloud infrastructures. As a whole SLAs are not only utilized to supply guaranteed service to end user, but are also utilized by providers to effectively supervise cloud infrastructures, by considering challenging priorities like energy efficiency while providing sufficient elasticity.

Due to the dynamic change in components such as workload and external conditions, hardware and software failures, already established SLAs may be violated. Continuous user interactions with the system during SLA negotiation and service executions at the time of failures might lead to gradual decrease in performance of Cloud Computing and this really arises the need for the development of SLA-aware Cloud.

3.1 SLA Aware Clouds

Guaranteed SLA of dynamic clouds mainly focus on elasticity which aims to meet QoS requirements such as performance and availability while minimizing cloud cost. In[11] the authors discussed the essential requirements for an SLA aware elastic cloud. These characteristics are:

Online observation and monitoring of the cloud
This is to periodically capture variations in cloud usage and workload to identify SLA violation and to generate cloud reconfiguration when necessary. QoS dimensions can be applied at different levels to offer low level metrics for IaaS clouds or higher level metrics for SaaS clouds. The main issue in this context is defining scalable, accurate and non intrusive distributed algorithms for cloud monitoring.

Modeling the cloud

A cloud has a vibrant performance with fluctuating and nonlinear cloud service work loads and this directly influence the quality (QoS) of cloud. A cloud is also categorized by its actual configuration (i.e. number of cloud services, location of machines hosting cloud services and service parameters of individual clouds) which influences both cloud QoS and cloud cost. Cloud modeling aims to concentrate the effect of workload, configuration on QoS and cost of the cloud service by defining a model which is precise and capable of accommodating the variation of cloud workload. This model is easy to use with real world applications.

Automated control of the cloud

Automated cloud control targets to build a dynamic elastic cloud that meets QoS requirements as specified in the SLA while reducing cloud cost. The use of a cloud model permits to target the variations of cloud configuration, workload and the corresponding effects on QoS and cost.

4 SLA and Resource Configuration

For configuring the resources in an efficient and sensible manner, a service provider has to consider several facts into account such as: Service Level Agreements, the Business Level Objectives of the service provider, the current status of the system and the job complexity of the system. In[8] the authors described the details about how far SLAs control the resource configurations and cloud services. Usually the terms in SLAs can not be utilized directly for configuring the affected resources. The Service Level Agreement may contain a collection of abstract terms which mean different concepts to different providers. As an example, for the term ”performance” different parties give different measures and so calculated and given in different ways according to the corresponding infrastructure. So for going towards the infrastructure layer from abstract terms, a mapping of high level terms to a low level is necessary which creates another agreement called Operational Level Agreement (OLA). A simple sample of SLA objectives (Table 1) and corresponding mapping rules (Table 2) are shown below for getting a clear idea of these parameters. Such a mapping is essential for a service provider because he has to be aware of what he wants to give to the service requesters to satisfy the terms like processing time, processing power etc. of an SLA.

Table 1. Table 1 SLA parameters

SLA Parameter	Possible value
Incoming bandwidth(bandwidthin)	>10 Mb/s
Outgoing bandwidth(bandwidthout)	>12 Mb/s
Storage(S)	1024 GB
Availability(A)	>99.2%
Response Time(R)	0.01ms

Table 2. Mapping of Resource metrics to SLA parameters

Resource Metrics	SLA parameter	Mapping Rules
Downtime,Uptime	Availability(A)	A=(1-(Downtime/Uptime))*100
Inbytes,outbytes,packetsize, bandwidthin, bandwidthout	Response Time(R)	R=Rin+Rout

The authors of [8] used an integrated approach which integrates infrastructure and business-specific knowledge to do autonomic translation of SLA parameters into configuring information. Such an autonomic adaptation process may perform the infrastructure management during runtime. SLAs have very high effect on configuring systems, particularly in case of dynamic on-demand service provisioning. This simplifies selection and formation of SLAs, by using a protocol and SLA representations and makes the usage of database to accommodate configuration information. But, studies of real business use cases revealed the fact that pre-defined configurations can only be used in a restricted category of scenarios, because the service providers system configuration does not only depend on the Service Level Agreements, but also on the job type(s) and the current workload.

Business Level Objectives such as *utilization of resources to a hundred percent* or *get the maximum profit while spending as little money as possible*, are not provided by available resource management systems. Additionally, a certain degree of knowledge of the service providers infrastructure is also required for the better usage of resource management. As a consolidation a configuration system that builds on a base configuration represented by BLOs and the complexity analysis of a job seems like a promising approach towards SLA-supported resource management.

5 Adaptive SLA Management

SLA management is to monitor the cloud service performance and to compare it with QoS parameters of the already established SLA. The result of this comparison can be logged in for later level of SLA negotiations and for billing of the usage of services. The SLA management framework manages the SLA violations by using the monitored information received from the monitor interfaces present in the framework. In adaptive SLA management certain level of autonomic control can be integrated to prevent the occurrence of an SLA violation by detecting the possibility of occurrence of a violation. Such a prior detection can be materialized by using an autonomic SLA management framework in association with the SLA negotiation strategy. The SLA negotiation strategy is basically meant for negotiating service performance and QoS parameters prior to the establishment of SLA and re-establishment of SLAs on detecting violations of SLA. Here in this proposed framework (shown in figure 1), the initial SLA establishment is done in the traditional way by negotiating the requirements of both the parties cloud service providers and cloud service customers. During the run, the SLA management framework monitors SLA parameters through its service monitoring interface and the possibility of an SLA violation will detect as an SLA violation threat.

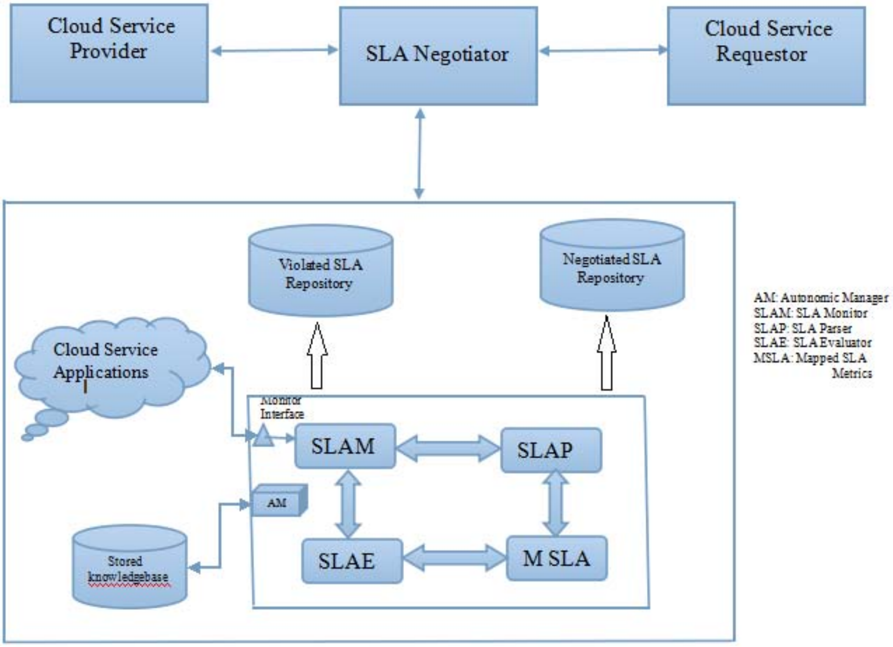


Fig. 1. Adaptive SLA management framework

The major blocks involved in this SLA management framework are SLA management, SLA parser, SLA Evaluator, Mapped SLA metrics and Autonomic Manager. The SLA management block (SLAM) contains multiple interfaces to monitor, negotiate, establish SLAs and focus on application management and self management of SLA adaptation. The main interfaces included here are: Monitor interface, Negotiation interface and Management interface. Monitor interface is equipped with several sensor elements which senses the variation in the system and convey the effects to negotiation interface. Negotiation interface do the necessary processing for the re-establishment of SLA agreements. Management interface is for providing management actions in order to reduce SLA violations by establishing necessary data transfers and control actions. The management interface use cloud services to sense changes in the desired state and to do some reactive actions for these sensed changes.

The SLA parser (SLAP) and Mapped SLA metrics (MSLA) are software modules intended for the parsing of SLA parameters and for establishing a mapping between resource metrics and SLA parameters. SLA Evaluator (SLAE) together with Autonomic Manager (AM) is for providing the necessary control to all other blocks and interfaces to establish adaptive SLA management.

On detection of an SLA violation threat, communications are given to both the resource provisioning unit and knowledge base repository. The resource provisioning unit will configure additional resources on seeing such a communication thereby the possibility of SLA violation may be avoided. The knowledge base repository is to store the details corresponding to each SLA violation and on the next level, these details are analyzed to calculate or modify the threshold limits. The results of these analysis is also forwarded to the SLA negotiation strategy for initiating the negotiation of re-establishment of SLAs after multiple SLA violations.

The detection of occurrence of SLA violation is done by using predefined threat thresholds which are more restrictive than violation thresholds. A violation threshold is a value which indicates the least acceptable performance level of the cloud service where as a threat threshold is a value generated from violation threshold by incorporating some reaction time to overcome SLA violations. Exceeding the threat threshold leads to a future SLA violation and this can be avoided by the quick reaction of the system thereby the cloud provider can save the SLA violation penalties. So the Business Level Objectives (BLOs) can also be maximized with the usage of this proposed SLA management framework.

Merits and Demerits

The merits of the newly proposed approach are, this mechanism will detect SLA violation well in advance and so it will be avoided either by allocating additional resources to the applications or by renegotiating the established SLA. So the cloud providers can save the unnecessary penalties for SLA violations. With the renegotiation of established SLAs the resource allocation will be done in a more efficient manner by limiting the over-provisioning of resources. The demerit in this case is, for keeping the knowledge base repository for SLA violation and Violation threat thresholds some extra memory resources are wasted.

6 Implementation Details

The proposed system is being studied for appropriate implementation and deployment using the ESPER[12] engine, GMOND module from the GANGLIA open source project[10] and Java Messaging Service (JMS) APIs. The processing and manipulation of SLA documents has done by Domain Specific Languages (DSLs) and the parameter

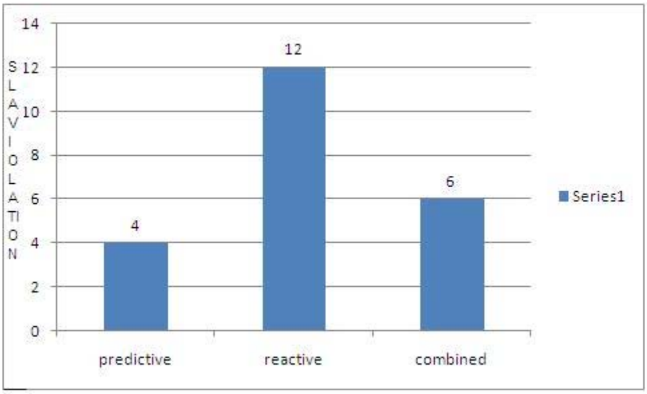


Fig. 2. Adaptive SLA management framework

extraction has done by XML parser. To implement the parameter mapping Java methods are used and the generated outputs will be forwarded to initiate the detection of SLA violations.

6.1 Preliminary Results

In the experimental set up we have created three virtual machines in two different physical machines and uploaded web applications on it. We have measured the quantity of predictive, reactive and combined modes of SLA violations over a fixed time span of 2 hours. Predictive mode means detection of the SLA violation case, reactive mode means avoidance of SLA violation and combined mode means the combination of both SLA detection and SLA avoidance. The obtained results are shown in figure 2.

7 Conclusion and Future Works

Flexible and reliable management of SLA agreements represents an open research issue in Cloud computing. Advantages of flexible and reliable Cloud infrastructures are manifold. Prevention of SLA violations avoid unnecessary penalties providers have to pay in case of violations. Timely reactions to possible SLA violations also reduces the need for human interactions. In this paper we proposed a new adaptive SLA management mechanism which can simultaneously focus on both the detection of occurrence of SLA violations and on the re-negotiation of established SLAs on multiple SLA violations. This management architecture consider an associated knowledge repository for storing the detected SLA violation threats and corresponding details. This stored details can be analyzed in the next level and the results of this analysis can be used for the negotiation of re-establishment of SLAs and for the modification/recalculation of SLA violation threat thresholds. With the usage of this newly proposed architecture the Business Level Objectives of cloud providers can also be improved to a maximum extent.

References

1. Boniface, M.J., Phillips, S.C., Sanchez-Macian, A., Surridge, M.: Dynamic service provisioning using GRIA sLAs. In: Di Nitto, E., Ripeanu, M. (eds.) ICSOC 2007. LNCS, vol. 4907, pp. 56–67. Springer, Heidelberg (2009)
2. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems 25(6), 599–616 (2009)
3. Comuzzi, M., Kotsokalis, C., Spanoudakis, G., Yahyapour, R.: Establishing and monitoring slas in complex service based systems. In: IEEE International Conference on Web Services, ICWS 2009, pp. 783–790. IEEE (2009)
4. Dobson, G., Sanchez-Macian, A.: Towards unified qos/sla ontologies. In: IEEE Services Computing Workshops, SCW 2006, pp. 169–174. IEEE (2006)
5. Emeakaroha, V.C., Brandic, I., Maurer, M., Dustdar, S.: Low level metrics to high level slas-lom2his framework: Bridging the gap between monitored metrics and sla parameters in cloud environments. In: 2010 International Conference on High Performance Computing and Simulation (HPCS), pp. 48–54. IEEE (2010)

6. Emeakaroha, V.C., Netto, M.A., Calheiros, R.N., Brandic, I., Buyya, R., De Rose, C.A.: Towards autonomic detection of sla violations in cloud infrastructures. Future Generation Computer Systems 28(7), 1017–1029 (2012)
7. Frutos, H.M., Kotsiopoulos, I.: Brein: Business objective driven reliable and intelligent grids for real business. IBIS 8, 39–41 (2009)
8. Hasselmeyer, P., Koller, B., Schubert, L., Wieder, P.: Towards SLA-supported resource management. In: Gerndt, M., Kranzlmüller, D. (eds.) HPCC 2006. LNCS, vol. 4208, pp. 743–752. Springer, Heidelberg (2006)
9. Koller, B., Schubert, L.: Towards autonomous sla management using a proxy-like approach. Multiagent and Grid Systems 3(3), 313–325 (2007)
10. Massie, M.L., Chun, B.N., Culler, D.E.: The ganglia distributed monitoring system: design, implementation, and experience. Parallel Computing 30(7), 817–840 (2004)
11. Redl, C., Breskovic, I., Brandic, I., Dustdar, S.: Automatic sla matching and provider selection in grid and cloud computing markets. In: Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing, pp. 85–94. IEEE Computer Society (2012)
12. Wood, T., Shenoy, P., Venkataramani, A., Yousif, M.: Sandpiper: Black-box and gray-box resource management for virtual machines. Computer Networks 53(17), 2923–2938 (2009)

Further Reading

1. Maurer, M., Breskovic, I., Emeakaroha, V.C., Brandic, I.: Revealing the mape loop for the autonomic management of cloud infrastructures. In: 2011 IEEE Symposium on Computers and Communications (ISCC), pp. 147–152. IEEE (2011)
2. Erdil, D.C.: Dependable autonomic cloud computing with information proxies. In: 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), pp. 1518–1524. IEEE (2011)
3. ESPER, Event Stream Processing Engine, `http://esper.codehaus.org/`
4. Foundation of Self-governing ICT Infrastructures (FoSII), `www.infosys.tuwien.ac.at/linksites/FOSII/index.html`
5. Brandic, I., Music, D., Dustdar, S.: Service mediation and negotiation bootstrapping as first achievements towards self-adaptable grid and cloud services. In: Proceedings of the 6th International Conference Industry Session on Grids Meets Autonomic Computing, pp. 1–8. ACM (2009)
6. Koller, B., Schubert, L.: Towards autonomous sla management using a proxy-like approach. Multiagent and Grid Systems 3(3), 313–325 (2007)
7. Brandic, I., Music, D., Leitner, P., Dustdar, S.: *vieSLAF* framework: Enabling adaptive and versatile SLA-management. In: Altmann, J., Buyya, R., Rana, O.F. (eds.) GECON 2009. LNCS, vol. 5745, pp. 60–73. Springer, Heidelberg (2009)
8. Java Messaging Service, `http://java.sun.com/products/jms/`

A Novel Audio Watermark Embedding and Extraction Method Based on Compressive Sensing, Sinusoidal Coding, Reduced SVD, Over Complete Dictionary and L1 Optimization

G. Jyothish Lal[1] and V.K. Veena[2]

[1] Karpagam Institute of Technology, Coimbatore, India
[2] Cognizant Technology Solution, Chennai, India
jyothishlal@gmail.com

Abstract. Digital audio watermarking is relatively a new technology to stop audio piracy and to ensure security of the ownership rights of the digital audio data. In this paper, a novel digital watermark embedding and extraction method for audio data is proposed, satisfying the demands of robustness and imperceptibility. This method is based on sinusoidal coding of speech, Compressive Sensing (CS), Reduced Singular Value Decomposition (RSVD), Over-Complete Dictionary (OCD) matrix and L_1 optimization algorithm. The sinusoidal approximation of original watermark signal is embedded into the compressive measurements of the host audio signal by using RSVD. Random sampling through compressive sensing ensures compression as well as encryption of the host audio signal. The extraction procedure is based on over-complete dictionary matrix and L_1 norm optimization. The over-complete dictionary is created by using sinusoidal speech coding bases and compressive sensing measurement matrix. Experimental results show that proposed method provide exact recovery of watermark information and host signal under noisy attacks.

Keywords: Audio Watermarking, Compressive Sensing (CS), Sinusoidal Speech Coding, Reduced SVD, Over-Complete Dictionary (OCD), L1 optimization.

1 Introduction

Nowadays, the digital multimedia technological advancement and decrease in prize of electronic gadgets like laptops, PCs, mobile phones etc made the distribution of digital data far easier. This is similar in the case of internet cost. People around the world can download and upload a large amount of multimedia data at a cheaper rate because of the tremendous advancement in broadband internet connections, 3G, 4G lite etc.

Besides having advantages like low quality degradation and cheap cost of recording device and distribution networks, digital media shows it weakness

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 160–171, 2014.

when it comes to matters like content authentication and copyright protection. That is, anyone can do unauthorized copying of these data if it is not copyright protected. This in turn results in false claiming of ownership right by a third party and a big financial crisis for the original copyright holders. So any illegal access should be prevented to stop piracy of digital multimedia data. Thus the demand for a technology that can preserve the security of the redundant multimedia data caused the evolution of digital watermarking. It is a unique technique to implement copyright information in multimedia data and thereby discourage digital data piracy [1]. The copyright information is normally hidden in the digital data to be transmitted through the insecure channel and it is later extracted to claim for the ownership rights. The type of watermark embedded into the digital media depends on the choice of application. This paper focus on the implementation of a robust digital watermarking, which includes new design strategies for watermark embedding and extraction in audio data and its subjective analysis.

Also, understanding of the human perception processes is the key to any successful watermarking schemes. So attention towards audio watermarking gave birth only after image and video watermarking schemes, because of the dynamic supremacy of Human Auditory System (HAS) over Human Visual System (HVS) [2, 3]. That is, the former has a wider dynamic range than the latter. So any design strategies for watermark embedding should preserve the imperceptibility of watermark embedded in the audio data. In the past decade, a number of audio data hiding or audio watermarking techniques in time domain and frequency domain [4–9] have evolved. Some other techniques [1, 3, 9] take credit of the perception and masking property of the human auditory system. That is, louder sounds acts as a masker for weaker sounds and human ear will not be able to perceive such sounds. The primary focus of any audio watermarking technique is to achieve perceptual transparency, high data rate and robustness together. But this is practically impossible. A maximum of two can be achieved in together by proper selection of cover medium and domain for embedding.

SVD has been used as an efficient technique in digital watermarking. Initially, it was developed for image watermarking [10–12]. Many of these methods manipulates the singular values for embedding the watermark information. This approach has the drawback that variation of largest singular values can be identified easily and are prone to suspicious attacks. Also, only a small number of singular values are available for manipulation, which in turn reduces the data payload. So, this paper deals with manipulating one of the unitary matrix of SVD decomposition, first proposed by [13], for embedding the watermark information. Here, prior to embedding the watermark, the digital audio data is compressed by using compressive sensing sampling. These compressive measurements are encrypted representation of digital audio data to be transmitted. In other words, Compressive sampling unifies the sampling, compression, and encryption of the digital audio data. This is attained by accumulating linear measurements $y = Ax$ of a sparse signal x where A is the linear transform carrying certain regularities. The linear measurements y are function of sensing matrix A [14]. Then, sinusoidal speech coding is done on the watermark audio signal to obtain sinusoidal bases. These sinusoidal approximations of watermark are embedded into the linear measurements taken for the host audio signal by using reduced SVD. Also, these bases together with the sensing matrix used in compressive sampling forms the over-complete dictionary. This OCD matrix is used for reconstruction of host audio signal and watermark audio signal separately by using L_1 optimization. A secret key is used for the creation of random measurement matrix at the encryption side, which is also shared with the decryption side. Therefore, the proposed method is a high sensitivity scheme since a slight variation in decryption key from the encryption key should not allow exact recovery of either the host audio signal or the watermark audio signal. Experimental results show that proposed system is more robust against noisy and compression attacks since an eavesdropper cannot perceive any intelligible information being transmitted over the insecure channel.

The rest of the paper is organized as follows. Section 2 gives a detailed picture about sinusoidal speech coding, compressive sensing theory, over-complete dictionary, reduced SVD and L_1 optimization. Section 3 describes the proposed embedding and extraction stage. Section 4 gives the experimental results , subjective and objective analysis. Finally section 5 concludes the paper.

2 Material and Methods

2.1 Sinusoidal Speech Coding

Fourier series can be used for representing a periodic signal. So this representation can be used since speech waveform has periodicity for some short duration of time, especially at the stage where vocal folds participates. Here, in speech waveform, periodicity changes regularly (say for every 30ms). Hence, it requires change of amplitude, fundamental frequency and phase for every few milliseconds. A voiced excitation can be decomposed into several harmonics, each one of which corresponds to a sine wave. On passing this sine wave excitation through a time varying vocal tract, it results in sine wave representation of the original speech waveform.

Due to the periodicity of speech waveform for a short frame, Short Time Fourier Transform (STFT) is taken for that frame and sinusoidal components are now considered as harmonic samples of the STFT. This can represented mathematically as

$$s(n) = \sum_{l=1}^{L} A_l \cos(nl\omega_0 + \phi_l) \ . \tag{1}$$

Here, sine wave frequency correspond to integer multiple of ω_0 . The Short Time Fourier Transform (STFT) of $s(n)$ is given by

$$S(w) = \sum_{n=-N/2}^{N/2} s(n)e^{-jnw} \ . \tag{2}$$

where N is the number of samples. The amplitude is estimated as $A_{l} = |S(lw_0)|$ and phase is estimated as $\phi_l = \arg S(lw_0)$ .The magnitude spectrum has peak at integer multiples of ω_0. Windowing techniques can be used for taking STFT. Usually, Hamming window is preferred over any another windows since it reduces side lobe leakage.

2.2 Compressive Sensing

Compressive Sensing (CS) or Compressive sampling is a new signal acquisition technique emerged in response to the demands of compressing and processing the increasing amount of data. Compressive Sensing theory relies on the sparsity of the signal of interest and tries to sense the signal from a fewer number of samples [15, 16]. That is, the signal is assumed to be sparse in some basis Ω and it can be concisely represented in that basis. Also, instead of sensing in a sparse representation, CS uses random measurements in a basis Φ that is incoherent with the sparse basis Ω. Incoherence means the signal which is sparse in one domain will be spread out in the other domain. Thus, universality principle is attained for CS as the same measurement technique can be utilized for signals that are sparse in different bases [17].

Suppose $x = \Omega\alpha$ is the interpretation of a real valued signal x of length N, where Ω is an an orthonormal basis matrix of size N×N and α is the vector of scalar coefficients of x. The Ω matrix gives K sparse representation of x with K<<N. This K sparse vector x is projected on to the random matrix Φ of size M×N to give M<N non-adaptive linear measurements y.

$$y = \Phi x = \Phi\Omega\alpha \ . \tag{3}$$

These random projections will preserve the information present in x with probability O(Klog(N/K)). In other words, as long as the rows in the matrix Φ grows linearly in K (ie; as sparsity grows, one should take more measurements), but logarithmically in N, the information in original signal can be preserved with high probability.

2.3 Over Complete Dictionary

A dictionary contains a set of elementary waveforms. The column vectors contained in a dictionary are basis functions which may be linearly dependent or independent and orthogonal or not orthogonal. A predefined dictionary Ψ of size N×M can be categorized into three based on the number of basis functions and length of the signal. That is, if M>N, Ψ is called as over complete and if M<N, Ψ is called as under complete and finally if M=N, it is complete. Here, the OCD matrix is constructed as follows:

$$\Psi = [\Phi\ D] \ . \tag{4}$$

where Φ is the random measurement matrix of size M×N and D are the sinusoidal bases of length N.

2.4 Reduced SVD and Embedding Algorithm

Singular value decomposition (SVD) is a most common technique for matrix decomposition. Initially, it was introduced for square matrix decomposition by Beltrami and Jordan in 1870s. Later, Eckart and Young extended it to rectangular matrices [18]. Full SVD of an M×N matrix A can be represented as the product of three matrices:

$$A = U\Sigma V^T \ . \tag{5}$$

where U and V are unitory matrix of size M×M and N×N respectively and Σ is the diagonal matrix containing non-negative elements or singular values. Compared to full SVD, reduced SVD is a much more compact and computationally efficient decomposition approach when N<<M. Thus,reduced SVD decomposes matrix A as $A = U\Sigma V^T$ where where U and V are unitory matrix of size M×R and R×N respectively (here, R is the rank of the matrix) and Σ is the diagonal matrix of size R×R with positive elements.

$$A = U\Sigma V^T = \begin{pmatrix} u_{11} & \dots & u_{1R} \\ \vdots & \ddots & \vdots \\ u_{M1} & \cdots & u_{MR} \end{pmatrix} \begin{pmatrix} \lambda_{11} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_{RR} \end{pmatrix} \begin{pmatrix} v_{11} & \dots & v_{1R} \\ \vdots & \ddots & \vdots \\ v_{N1} & \cdots & v_{NR} \end{pmatrix}^T \ . \tag{6}$$

In this paper, instead of the Σ matrix, unitory matrix U is used for embedding the sinusoidal approximation s_n of the watermark. The algorithm is as follows:

$$\left.\begin{array}{l} A = U\Sigma V^T \\ U_w = U + \beta W \\ A_w = U_w \Sigma V^T \end{array}\right\} \ . \tag{7}$$

where W is the matrix of sinusoidal approximation of watermark and β is any scalar. This A_w is then transformed to 1-D signal which is the watermarked audio signal Y.

2.5 Signal Reconstruction via L$_1$ Optimization

From the watermarked signal Y, which contains the non-adaptive linear measurements and sinusoidal approximation, the host audio signal and watermark audio frame are recovered as follows:

$$\arg\min_{\theta} \|\theta\|_1 \quad \textit{subject to } Y = \Psi\theta \ \textit{ where } \theta = \begin{bmatrix} x \\ s_n \end{bmatrix} \ . \tag{8}$$

This is convex optimization problem and it is solved based on L$_1$ regularized formulation [19, 20].

$$\tilde{\theta} = \arg\min_{\theta} \left\{ \|\Psi\theta - Y\|_2^2 + \lambda \|\theta\|_{\ 1} \right\} \ . \tag{9}$$

where λ is the regularization parameter.

3 Proposed Method

3.1 Embedding Stage

In the embedding stage, the host audio data is acquired or sensed via compressive sampling. Fig. 1 shows the proposed embedding stage.

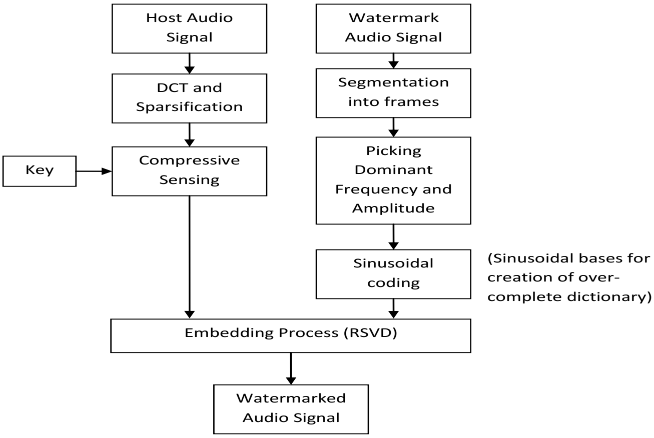


Fig. 1. Embedding stage

Since Compressive Sensing (CS) relies on sparsity of signal, the host signal initially passes a sparsification process and then CS is performed by a linear measurement step, by creating a measurement matrix. The generation of random measurement matrix uses a secret key (shared with the extraction stage). After compressive sampling, the host audio signal look like noise signal and mostly attackers will ignore it. Then, these encrypted measurements are converted to a matrix format. Now, the watermark audio signal is split into frames of length N. The magnitude spectrum of each of these frames is computed to pick out the dominant amplitude, frequency and phase. These spectral components are then used for making the sinusoidal approximation of the watermark audio signal. Now, for applying RSVD, the sinusoidal bases from each of the frames are arranged into a matrix format. Then, both encrypted measurements and sinusoidal approximation are embedded and transomed into 1-D audio signal to form the watermarked audio signal.

3.2 Extraction Stage

The extraction stage reads the over-complete dictionary matrix and solve the L_1-norm minimization problem. The coefficients obtained after L_1 optimization are manipulated to separate host audio signal and watermark audio signal. That is, here, the first N coefficients correspond to host audio signal and N+1 to end coefficients correspond to watermark audio signal. Also, decryption of the host audio signal requires an inverse DCT operation. Fig. 2 shows the proposed extraction stage.

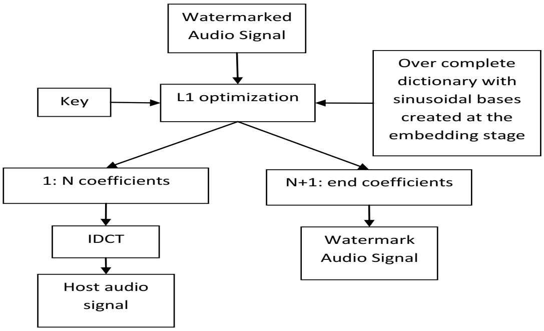


Fig. 2. Extraction stage

4 Experimental Results

For evaluation of the proposed embedding and extraction method, a host audio signal sampled at 11025 Hz and watermark audio signal sampled at 10000 Hz were used. Initially, the host audio signal with 6027 samples undergoes a sparsification process. That is, the Discrete Cosine Transform (DCT) for the host audio signal is calculated and coefficients which does not correspond to the signal intelligence are discarded by passing through a gateway. The threshold limits for the gateway used here are 0.035 (upper) and -0.075 (lower). Fig.3 shows the sparsified DCT spectrum for the host audio data.

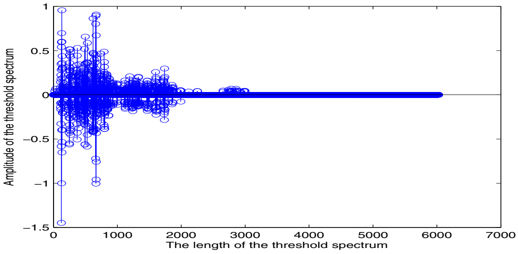


Fig. 3. Sparsified DCT spectrum of host audio data

Then Compressive Sensing is applied to the sparse vector by taking random measurements out of it. Here, 1024 measurements were taken. Fig. 4 shows the CS output of the host audio data, which are encrypted representation of the same.

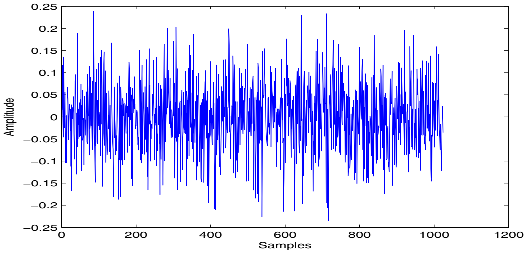


Fig. 4. CS output of the host audio data

The frames for embedding are prepared by passing the watermark audio signal through a hamming window of length 1024. Fig. 5 shows the hamming window used.

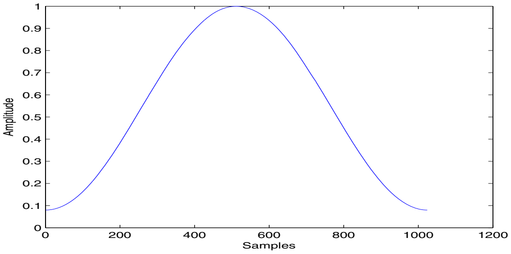


Fig. 5. Hamming window of length 1024

Fig. 6 shows the one frame of watermark audio signal. Then, Fourier transform of the test frame is taken to identify the frequency components present in it. Fig. 7 shows the single sided amplitude spectrum of the test frame of the watermark audio signal.These spectral components are used for creating sinusoidal bases, which are now called as the watermark information. Here, six sinusoidal bases are created for the test frame used. Now, both the encrypted measurements and sinusoidal approximations are organized into 128×8 matrices. Reduced SVD is applied to matrix of encrypted measurements and three matrices U, Σ and V are derived. Then watermark information is embedded into the columns of U matrix.

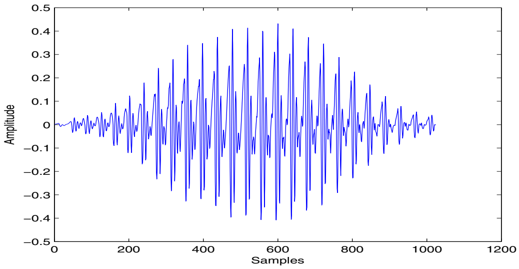


Fig. 6. One frame of watermark audio signal

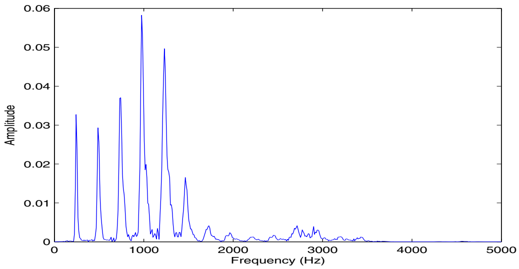


Fig. 7. Single sided amplitude spectrum of the test frame of the watermark audio signal

Fig. 8 shows the watermarked audio signal. In the extraction stage, L_1 norm minimization procedure solves the problem of size 1024×6033. The regularization parameter is fixed at 0.01. Then, first 6027 coefficients obtained after L_1 optimization undergoes inverse DCT operation to obtain the host audio signal.

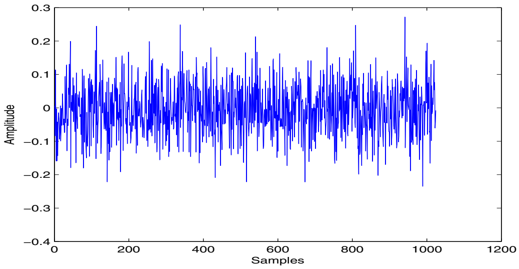


Fig. 8. Watermarked audio file

Coefficients from 6028 to 6033 gives watermark audio frame. Fig. 9 and Fig. 10 shows the reconstructed watermark audio frame and host audio signal respectively.

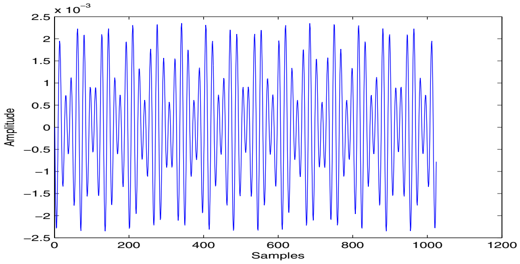


Fig. 9. Reconstructed watermark audio frame

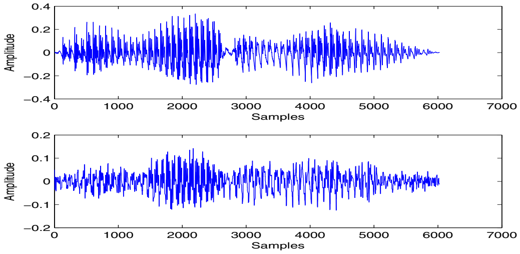


Fig. 10. Original and reconstructed host audio data

4.1 Subjective and Objective Analysis

The performance of the proposed method is measured via subjective and objective means. Subjective analysis of the extracted watermark signal and host signal were also conducted. Five listeners are provided with original host audio signal and watermark audio frame to identify the dissimilarity of the same with reconstructed ones and put their grades accordingly, for selected attacks. Average of the subjective grades is taken as Mean Opinion Score (MOS) for the proposed method. Objective analysis is performed by calculating the Normalized Correlation (NC) and Signal to Noise Ratio (SNR) between the original and extracted audio files for selected attacks.

$$\mathrm{SNR} = 20\log_{10}\left(\frac{\|F\|^2}{\left\|F-\tilde{F}\right\|^2}\right) . \tag{10}$$

$$NC(F,\tilde{F}) = \left\langle \frac{F}{\|F\|}, \frac{\tilde{F}}{\left\|\tilde{F}\right\|} \right\rangle . \tag{11}$$

where F and $\tilde{F}$ are the original and extracted audio files, $\langle\cdot,\cdot\rangle$ is the inner product and $\|\cdot\|$ is the L^2 norm. Table 1 list out the NC, SNR values and MOS grades obtained for watermark signal and host signal after various attacks.

Table 1. Robustness Evaluation Results(W denotes watermark, H denotes Host)

Attack Type	NC(W,H)	SNR in dB(W,H)	MOS grade(W,H)
No attack	1, 1	29.83, 30.1	4.30, 4.35
MP3 compression	0.9944, 0.9920	28.86, 27.53	4.10, 4.20
Gaussian Noise	0.9996, 0.9998	28.12, 30.01	4.20, 4.25
Low pass Filtering	0.9981, 0.9872	28.56, 29.89	4.00, 4.10
Resampling	0.9995, 0.9944	27.28, 27.56	4.00, 4.20
Denoising	1, 0.9997	29.87, 28.91	4.20, 4.25
Cropping	0.9920, 0.9850	28.50, 27.55	4.10, 4.20

5 Conclusion

Audio watermarking seems a flawed concept as and when it was introduced into the digital watermarking arena. Whether the watermark is inaudible or audible, attacks like compression, low pass filtering etc can adversely affect the watermark inserted and thus it will be difficult to claim the ownership rights. So this paper focus on the security as well as efficient reconstruction of the watermark information. Compressive sampling provides efficient encryption besides compression of the host audio data and sinusoidal coding helps to represent watermark information as sum of sinusoids. The method has the advantage that, inherently the watermark information is saved in the over-complete dictionary. The use of reduced SVD and watermark embedding in unitary matrix gives more data capacity as well as computational efficiency for the proposed method. The method also give wrong impression about the watermarked signal for an attacker, as it will be misinterpreted as noise signal when transmitted over the insecure channel. The reconstruction procedure is more compromising as it facilitates exact recovery of the host audio signal and watermark audio signal. Thus the proposed method is an efficient watermarking scheme for the prevention of illegal copying and manipulation of audio files and it fulfils the demands of imperceptibility and robustness to a great extend.

References

1. Podilchuk, C.I., Delp, E.J.: Digital Watermarkin- Algorithms and Applications. IEEE Signal Processing Magazine, 33–46 (July 2001)
2. Wang, X.-Y., Zhao, H.: A Blind Audio Watermarking Algorithm Robust Against Synchronization Attack. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-M., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005. LNCS (LNAI), vol. 3802, pp. 617–622. Springer, Heidelberg (2005)
3. Kim, Y.H., Kang Il, H., Kim Il, K., Han, S.-S.: A Digital Audio Watermarking Using Two Masking Effects. In: Chen, Y.-C., Chang, L.-W., Hsu, C.-T. (eds.) PCM 2002. LNCS, vol. 2532, pp. 655–662. Springer, Heidelberg (2002)
4. Lemma, A.N., Aprea, J., Oomen, W., VandeKerkhof, L.: A temporal domain audio watermarking technique. IEEE Trans. Signal Process. 51(4), 1088–1097 (2003)

5. Lie, W.N., Chang, L.C.: Robust high-quality time-domain audiowatermarking based on low-frequency amplitude modification. IEEE Trans. Multimedia 8(1), 46–59 (2006)
6. Kirovski, D., Malvar, H.S.: Spread-spectrum watermarking of audio signals. IEEE Trans. Signal Process. 51(4), 1020–1033 (2003)
7. Wu, S., Huang, J., Huang, D., Shi, Y.Q.: Efficiently self-synchronized audio watermarking for assured audio data transmission. IEEE Trans. Broadcast. 51(1), 69–76 (2005)
8. Wang, X.Y., Zhao, H.: A novel synchronization invariant audio watermarking scheme based on DWT and DCT. IEEE Trans.Signal Process. 54(12), 4835–4840 (2006)
9. Blackledge, J., Omar, F.: Audio data verification and authentication using frequency modulation based watermarking. International society for advanced science and technology. J. Electron. Signal Process. 3(2), 51–63 (2008)
10. Chung, K., Yang, W., Huang, Y., Wu, S., Hsu, Y.: On SVD-based watermarking algorithm. Appl. Math. Comput. 188(1), 54–57 (2007)
11. Hu, Y., Chen, Z.: An SVD-based watermarking method for image authentication. In: Proceedings of International Conference on Machine Learning and Cybernetics, vol. 30, pp. 1723–1728 (August 2007)
12. Liu, R.Z., Tan, T.N.: An SVD-Based Watermarking Scheme for Protecting Rightful Ownership. IEEE Trans. on Multimedia 4(1), 121–128 (2002)
13. Wangn, J., Healy, R., Timoney, J.: A robust audio watermarking scheme based on reduced singular value decomposition and distortion removal. J. SignalProcessing 91, 1693–1708 (2011)
14. Candes, E., Wakin, M.: An introduction to compressive sampling- A sensing paradigm that goes against the common knowledge in data acquisition. IEEE Sig. Proc. Mag. 25(2), 21–30 (2008)
15. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory 52(4), 1289–1306 (2006)
16. Orsdemir, A., Oktay Altun, H., Sharma, G., Mark Bocko, F.: On the security and robustness of encryption via compressed sensing. In: Proc. IEEE Military Communications Conference, MILCOM 2008, pp. 1–7 (November 2008)
17. Griffin, A., Hirvonen, T., Tzagkarakis, C., Mouchtaris, A., Tsakalides, P.: Single-Channel and Multi-Channel Sinusoidal Audio Coding Using Compressed Sensing. IEEE Trans. on Audio, Speech and Language Processing 19(5), 1382–1395 (2011)
18. Johnson, R.M.: On a theorem stated by Eckart and Young. Psychometrika 28(3), 259–263 (1963)
19. Candes, E., Romberg, J.: l_1-magic : Recovery of Sparse Signals. Caltech, via Convex Programming (October 2005)
20. Tropp, J.A., Gilbert, A.C.: Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory 52(12), 4655–4666 (2007)

Compressive Sensing Based Audio Scrambling Using Arnold Transform

Nishanth Augustine, Sudhish N. George, and P.P. Deepthi

Department of Electronics and Communication
National Institute of technology, Calicut

Abstract. In this paper, a novel idea for scrambling the compressive sensed audio data using two dimensional Arnold transform is presented. In the proposed method, Arnold matrix is constructed by the numbers generated by using a secret key and a logistic map. A key based measurement matrix is used for compressive sensing to avoid the transmission and storage requirement of the matrix and to improve the security. The combination of compressive sensing and arnold scrambling provides very high security and ensures efficient channel usage, resistivity to noise, best signal to noise ratio and good scrambling of data. Experimental results confirm the effectiveness of the proposed scheme.

1 Introduction

Scrambling is a technique which is mainly used in data hiding, watermarking and encryption applications for providing information security against illegal surveillance and wire tapping. In the time domain scrambling process, a segment of time domain sample values are taken and scrambles them into a different segment of samples. At the receiving end, the scramled data is descrambled into its original form. Both the scrambling and descrambling operations are based on a scrambling matrix. The disadvantage of audio scrambling matrices constructed by pseudorandom sequences [6], Hadamard transform [10] and Fibonacci transform [8] is that, since these matrices are invariable, they could easily be deciphered. Some improved algorithms such as stochastic matrix [5] and latin square [9] were developed to overcome this problem, but they result in heavy transmission load. Speech compression methods like G.729 mixed excitation linear prediction (MELP) and adaptive multi-rate (AMR) [11] audio codec are then employed along with the process of scrambling to reduce the transmission load, but these methods shows low robustness in the presence of noise.

The degree of security of a scrambling algorithm depends on residual intelligibility and key space [2]. Residual intelligibility is the amount of intelligibility left over in the scrambled signal. The lower the residual intelligibility of a scrambling method, the higher its degree of security. Scrambling degree (SD) [7] can be used to evaluate the degree of security. As SD increases, degree of security increases and residual intelligibility decreases. Key space is the number of keys available for scrambling. Larger the key space better will be the degree of security.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 172–183, 2014.

An efficient scrambling method should be channel-saving, attack-resistant and should provide high scrambling degree. Since compressive sensing (CS) [3] provides very good compression and robustness whereas Arnold scrambling [12] provides very good scrambling degree, by combining both these techniques, an effective audio scrambling method can be developed. In the proposed scheme, compressive sensing is applied on the audio signal and the resultant lower dimensinal vector is scrambled using Arnold transform.

Compressive sensing performs both sampling as well as compression, along with encryption of the source information simultaneously. CS seeks to represent a signal using a number of linear, non-adaptive measurements. Usually, the number of measurements is much lower than the number of samples needed if the signal is sampled at the Nyquist rate. CS requires that the signal is sparse in some basis and it combines the steps of sampling and compression. CS stores and transmits only a few non-zero coefficients, and then enables recovery of signals from them. This greatly reduce the time of data acquisition, storage and the amount of data needed to be transmitted. This is the prominent advantage of CS [3]. The random meaurements are taken by using a measurement matrix of suitable size and the receiver should know this matrix for the reconstruction. A key based measurement matrix not only provides security, but also eliminates the necessity of transmission and storage of the same.

Arnold Transform is a transformation technique which is mainly used in image scrambling to rearrange the pixels of the image randomly [12]. A two dimensional Arnold transform can also be used for scrambling audio data, since it breaks the correlation between audio samples effectively. An algorithm based on a 64 bit key and a logistic map is used for constructing the Arnold matrix which is used for rearranging the data [4]. Arnold scrambling offers excellent scrambling degree.

The rest of the paper is organized as follows. Section 2 illustrates the basics of compressive sensing. Section 3 covers the Arnold transformation algorithm. The proposed scrambling scheme is discussed in Section 4. Section 5 gives the analysis and discussion of the experimental results. Conclusions are drawn in section 6.

2 Compressive Sensing

Compressive sensing, also known as compressive sampling [3], is an emerging field which relies on the sparsity of the signal. By employing CS, a great majority of the data can be compressed by sampling the signal at Sub-Nyquist rate (also called subrate), which is much lower than the Nyquist rate. CS theory is based on the assumption that the signal of interest is sparse in some basis as it can be accurately and efficiently represented in that basis.

The basic idea behind CS is that the signals that are composed as linear combinations of few linearly independent vectors need only to be sampled at a low rate to facilitate a high quality reconstruction [3]. Here, few means that the number of basis vectors is small relative to the number of samples. More specifically, if a signal is composed of a linear combination of T vectors, we can reconstruct the signal using kT samples formed as random linear combinations of the N original samples, where k is a small positive integer. It is then clear that if kT is much smaller than N, we have achieved a compression provided the level of sparsity is known a priori. Compressive sensing uses random measurements in a basis that is incoherent with the sparse basis. Incoherence [13] means that no element of one basis has a sparse representation in terms of the other basis. CS has found applications in many areas such as image processing, spatial localization, medical signal processing etc.. In addition, CS is particularly suited to multiple sensor scenarios, making it a good choice for wireless sensor networks.

2.1 Measurement and Reconstruction

Consider a signal X which is an $N \times 1$ vector. Let X be sparse in some another domain Ψ such that

$$X = \Psi v \tag{1}$$

where v is the sparse coefficients of X which is an $N \times 1$ vector, but with only T coefficients ($T << N$) are non zero. Ψ is called dictionary matrix and of size $N \times N$, i.e. X can be represented by using only T coefficients in Ψ. The main idea of CS is to map the observed signal X to a lower-dimensional vector Y via measurement matrix Φ by the following transformation:

$$Y = \Phi X \tag{2}$$

where Φ is an $M \times N$ matrix (T < M < N) and should satisfy restricted isometric property (RIP) [1] and should be incoherent with Ψ.

To do reconstruction, we need to introduce a sparsity metric on v. A commonly used measure for this is the l_1- norm [1], denoted as $||.||_1$, which can be related to the number of non-zero coefficients under certain technical conditions. The vector v is said to be T sparse if it contains only T non-zero coefficients. We can now pose the sparse decomposition problem as the following:

$$minimize\ \ ||v||_1\ \ s.t.\ \ X = \Psi v \tag{3}$$

Similarly, the measurement matrix Φ is also multiplied on to Ψv, and we can write the sparse decomposition problem as

$$minimize\ \ ||v||_1\ \ s.t.\ \ Y = \Phi\Psi v \tag{4}$$

introducing $\Theta = \Phi\Psi$, above problem can be rewritten as

$$minimize\ \ ||v||_1\ \ s.t.\ \ Y = \Theta v \tag{5}$$

The key point of CS theory is that by using an appropriate measurement matrix Φ, the solution of above equation will result in a vector v' which will reconstruct not only Y, but also X exactly as $X = \Psi v'$ when v is sparse [13].

Above equation can be solved either by employing a convex optimization technique or by using a greedy algorithm. The convex optimization method provides robustness against noise and guarantees the reconstruction of all sparse signals, but it lacks the speed of greedy approach. On the other hand, the greedy methods had not been able to provide the strong guarantees of convex optimization method, but they are faster [13].

3 Arnold Transform

Arnold transformation is commonly known as cat face transformation and is actually a location moving of a point. Suppose (x, y) is a point in a matrix of size $p \times q$, then the transformation that change the point (x, y) to another point (x', y') is given by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \left(\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \right) mod \begin{bmatrix} p \\ q \end{bmatrix}$$

This transformation is called two dimensional Arnold transformation [12]. This is an equeal area transformation and it can be iterated. Arnold transformation is cyclical, that is, when iterate to a certain step, it will regain the original location. Since there are many methods for calculating the periodicity and getting the inverse transformation, the use of traditional Arnold transformation for the scrambling has become unsafe. So the traditional Arnold transformation is modified by adding two parameters a and b, where a and b are positive integers and the transformation is as below.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \left(\begin{bmatrix} 1 & a \\ b & ab + 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \right) mod \begin{bmatrix} p \\ q \end{bmatrix} \tag{6}$$

We can choose different transform coefficient a and b and it is difficult to regain the original location after the transform because the transform coefficient is not the only, which can improve the efficiency of scrambling algorithm and security [12].

4 Proposed Scheme

The proposed scrambling scheme has to perform two processes: compressive sensing and scrambling. First the audio signal is compressed by taking random measurements of original samples, and then the compressed samples are scrambled using Arnold transform. The scrambling algorithm takes the compressed file as the input and produces a scrambled output. At the receiver side this file is descrambled and the original file is reconstructed from the descrambled data. Two methods are used for the reconstruction: one convex optimization algorithm, l_1- minimisation [13] and one greedy approach, the regularized orthogonal matching pursuit (ROMP) [13].

4.1 Arnold Matrix Generation

The Arnold matrix, used for scrambling is constructed by using a logistic map and a 64-bit key. The logistic map is a polynomial mapping which can be expressed mathematically as $x_{n+1} = \mu x_n[1 - x_n]$, where x_n is a number between zero and one and μ is a positive number. A logistic map exhibit a great sensitivity to initial conditions if the values of μ is in between about 3.57 and 4 and hence it can be considered as a chaotic system. The logistic map scheme is reliable, unpredictable, offers randomness and does not require any computationally intensive algorithms. The steps for Arnold matrix generation are given below [4]:

1. Divide the 64-bit key into two 32-bit binary numbers and then find their decimal equivalents, K_1 and K_2 and their sum K_T.
2. Calculate $I_1 = K_1/K_T$ and $I_2 = K_2/K_T$.
3. Iterate I_1 and I_2, T times using the logistic map $I_j(i+1)= \mu I_j(i)[1 - I_j(i)]$ for $j = 1, 2$ and $i = 1, 2...T$.
4. Identify the most significant three bits, $b1_j, b2_j, b3_j$ of I_j for $j = 1, 2$.
5. Generate two decimal numbers a_1 and a_2 using the equation $a_j = 100b1_j + 10b2_j + b3_j$ for $j = 1, 2$ and calculate their product a_3.
6. Construct Arnold matrix, A using the above values as

$$A = \begin{bmatrix} 1 & a_1 \\ a_2 & a_3 + 1 \end{bmatrix}$$

4.2 Scrambling Algorithm

The procedure starts by reading an audio file and determining its length, N. Choose suitable subrate, S (subrate is the ratio between the length of compressed file to that of original file) and calculate the number of random measurements required, M ($M = N \times S$). The size of the transformed file is reduced to M by taking random measurements, i.e. by multiplying with a measurement matrix of size $M \times N$ which is composed of numbers, generated randomly based on a 32- bit key. Once the signal has gone through the process of CS then scramble it using Arnold transformation. For that reshape the 1D measurement vector to a 2D array having M cells. Construct the 2D Arnold matrix by the numbers generated by using the 64-bit key and logistic map. The new indices values used for scrambling is obtained by multiplying the current indices values with the Arnold matrix, K times. The compressed audio samples are transformed into another two dimensional array according to this indices list. After filling all audio samples, the two dimensional matrix is converted into one dimensional array. This scrambled audio file is written with the same sample rate and number of bits per sample as its original. The algorithm can be described as follows:

1. Read audio file X and determine its length N.
2. Fix a subrate S and calculate M.
3. Construct the key based measurement matrix Φ of size $M \times N$.

4. Take M random measurements of X' by multiplying it with the measurement matrix Φ.
5. Reshape the measurement vector Y into a rectangular matrix $Y1$ of size $p \times q$, where $p \times q = M$.
6. Calculate the new index values, (x', y') by multipying the current index values, (x, y) with Arnold matrix, using equation(6) A for $x = 1...p$ and $y = 1...q$.
7. Repeat above step K times.
8. Construct another matrix $Y2$ such taht $Y2(x', y') = Y1(x, y)$.
9. Reshape $Y2$ to a 1D sequence Z of size $M \times 1$.
10. Write the scrambled file Z in the same format, with the same sample rate and number of bits per sample as that of X.

In this scrambling method, the degree of security relies on two keys; a 32-bit key for the measurement matrix generation and a 64-bit key for the Arnold matrix generation. Thus the key space is 2^{96}, which is enough for preventing brute force attack. Key space can be further increased by keeping the parameters μ, T and K as secret. The use of CS and key based matrix generation prevents an attacker from the inverse mapping of scrambled data even if he has some knowledge about the plain text. Thus this scheme provides security against known plain text attack.

4.3 Descrambling Algorithm

It takes the scrambled file Z, descramble it and original audio file is reconstructed from this descrambled file. The descrambling process is similar to that of scrambling process. The only difference is that the descrambling matrix is the inverse of Arnold matrx, A. The audio file, X_{rec} can be reconstructed by applying l_1- minimisation or ROMP to the descrambled file.

5 Experimental Results

Experimental results shows that in addition to being robust to data loss attacks, the proposed scheme can provide very high security by breaking the correlation between audio samples effectively and can reduce the transmission load considerably. This algorithm is applicable to speech and music audio files having different sizes.

The performance of the proposed scrambling scheme is evaluated from four perspectives :

(1) The scrambling degree, (2) Correlation coefficient, (3) The reconstruction quality and (4) Resistance to noise. We test several audio files for different values of subrate S by varying it from 0.1 to 0.5. In Section 5.1 scrambling degree test results and discussions are presented. Correlation analysis were presented in Section 5.2. Section 5.2 verifies the reconstruction quality and the result of verification of the robustness to noise is given in Section 5.4.

5.1 The Scrambling Degree SD

Scrambling degree (SD) [7] is a measure which is used to indicate the performance of scrambling algorithm. It can be calculated as follows:

Let $P(i)$ be the original audio sample and L is the length of the audio file, then the difference D for i^{th} cell is calculated as follows:

$$D(i) = \frac{1}{4} \sum_{i'} P(i) - P(i') \tag{7}$$

where $(i') = [(i-1), (i-2), (i+1), (i+2)]$
Then the mean difference M for the audio file is calculated as

$$M = \frac{\sum_{i=3}^{L-2} D(i)}{L-4} \tag{8}$$

The scrambling degree SD is defined as

$$SD = \frac{M' - M}{M' + M} \tag{9}$$

where M' is the mean difference of the audio file, reconstructed after compressive sensing and scrambling, i.e. without performing descrambling and M is the mean difference of the original audio file. The values of SD ranges from -1 to 1. Higher value of SD indicates better scrambling.

The scrambling degree for different values of subrate S is tested by using different files of different size. The results obtained are shown in Table 1. From these results it is evident that the proposed method guarantees excellent scrambling performance. As subrate S increases SD also increases and approaches its maximum value.

Table 1. SD values of different files for various subrate, S

Sl.No.	File	S=0.1	S=0.2	S=0.3	S=0.4	S=0.5
1	music	0.9368	0.9538	-0.9547	-0.9562	-0.9569
2	voice	0.8585	0.8880	0.8896	-0.8902	-0.8973
3	speech	0.9408	-0.9522	-0.9533	-0.9538	-0.9541
4	mix	0.8799	0.9106	-0.9132	-0.9156	-0.9184
5	guitar	0.9623	0.9716	0.9730	0.9762	0.9794

5.2 Correlation Coefficient

Correlation coefficient, denoted by ρ is a measure of similarity of two waveforms, giving a value between +1 and -1 inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is negative correlation. It is widely used as a measure of the degree of linear dependence between two variables and is defined as the covariance of the two waveforms divided by the product of their standard deviations. Mathematically it can be expressed as

$$\rho = \frac{cov(X, Xrec)}{\sigma_X \sigma_{Xrec}} \tag{10}$$

where $cov(X, Xrec)$ is the covariance of original and reconstructed audio files and σ_X and σ_{Xrec} are the standard deviations of orifinal and reconstructed audio files respectively.The results of correlation analysis of original audio signal to the reconstructed signal are given in Table 2 and 3. Nearly perfect correlation is obtained for all the files, if it is reconstructed after performing both scrambling and descrambling. The files reconstructed using scrambled files shows poor correlation to the original files. From this analysis it is evident that this scrambling scheme breaks the correlation between audio files excellently.

Table 2. Correlation coefficient of different files (correct reconstruction) for various subrate, S

Sl.No.	File	S=0.1	S=0.2	S=0.3	S=0.4	S=0.5
1	music	0.9327	0.9735	0.9865	0.9923	0.9953
2	voice	0.6291	0.8795	0.9582	0.9819	0.9918
3	speech	0.8703	0.9636	0.9836	0.9933	0.9954
4	mix	0.7632	0.8829	0.9335	0.9602	0.9745
5	guitar	0.9244	0.9827	0.9937	0.9968	0.9981

Table 3. Correlation coefficient of different files (reconstructed using scrambled file) for various subrate, S

Sl.No.	File	S=0.1	S=0.2	S=0.3	S=0.4	S=0.5
1	music	0.0007	0.0026	0.0013	-0.0007	-0.0014
2	voice	0.0009	-0.0038	-0.0062	-0.0011	-0.0008
3	speech	-0.0003	0.0023	-0.0035	0.0077	-0.0042
4	mix	0.0002	-0.0005	-0.0028	-0.0016	-0.0009
5	guitar	0.0021	0.0020	-0.0001	0.0015	0.0011

5.3 The Reconstruction Quality

The reconstruction quality can be measured by calculating signal to noise ratio (SNR). It is defined as the ratio between original signal power to the noise power. Noise is termed as the difference between the original audio sample values and the reconstructed sample values. SNR can be calculated as

$$SNR = 10 \log_{10} \frac{X^2}{(X - Xrec)^2} \tag{11}$$

where $Xrec$ is the reconstructed audio file.

SNR of different files, reconstructed by using l_1- minimisation and ROMP, after scrambling and descrambling, for different subrate is shown in Table 4 and Table 5 recpectively. From these results it is clear that the reconstruction quality of both algorithms is good. At lower values of subrate S, ROMP is performing better than l_1- minimisation. But as S increases, performance of l_1- minimisation is superior to that of ROMP. In all cases SNR increases with S.

Table 4. SNR values of different files for various subrate, S (l_1- minimisation)

Sl.No.	**File**	**S=0.1**	**S=0.2**	**S=0.3**	**S=0.4**	**S=0.5**
1	music	8.0104	11.9917	15.4787	18.6729	22.1571
2	voice	2.6756	7.02010	12.3697	20.0540	33.7731
3	speech	6.3885	11.6741	16.2038	19.9260	23.3911
4	mix	3.0673	6.25460	9.07000	11.7339	14.5620
5	guitar	7.9383	13.9967	19.0281	23.0887	26.0645

Table 5. SNR values of different files for various subrate, S(ROMP)

Sl.No.	**File**	**S=0.1**	**S=0.2**	**S=0.3**	**S=0.4**	**S=0.5**
1	music	8.8274	12.7855	15.7081	18.1484	20.2343
2	voice	2.7670	6.3381	10.8529	14.4348	17.8786
3	speech	6.0815	11.4175	14.8611	16.3508	18.6464
4	mix	3.5971	6.4153	8.8326	11.0446	12.9644
5	guitar	8.3503	14.6294	19.0014	21.9538	24.3163

The effect of scrambling can also be verified by reconstructing the audio signal using the scrambled file, i.e. without performing descrambling and then calculating SNR using this reconstructed signal. The result is shown in Table 6 and Table 7, and it clearly shows that SNR is very poor for all values of S.

Table 6. SNR values of different files (reconstructed using scrambled file) for various subrate S (l_1- minimisation)

Sl.No.	**File**	**S=0.1**	**S=0.2**	**S=0.3**	**S=0.4**	**S=0.5**
1	music	-1.4787	-1.8100	-2.0463	-2.2347	-2.4043
2	voice	-1.4577	-1.8037	-2.0426	-2.2464	-2.3616
3	speech	-1.4993	-1.8385	-2.0561	-2.2163	-2.4191
4	mix	-1.4713	-1.8124	-2.0686	-2.2413	-2.4413
5	guitar	-1.4921	-1.7984	-2.0616	-2.2369	-2.4369

Table 7. SNR values of different files (reconstructed using scrambled file) for various subrate, S (ROMP)

Sl.No.	File	S=0.1	S=0.2	S=0.3	S=0.4	S=0.5
1	music	-2.2806	-2.8971	-2.9122	-2.9347	-2.9901
2	voice	-2.3832	-2.9927	-3.0281	-3.1464	-3.3616
3	speech	-2.3579	-2.8417	-2.9131	-2.9202	-2.9314
4	mix	-2.3326	-2.9147	-2.9968	-3.0211	-3.1134
5	guitar	-2.2727	-2.8677	-2.9781	-2.9963	-3.0812

The following plots shows the effectiveness of the proposed algorithm. Original signal is displayed in Fig. 1. Reconstructed signal using scrambling and descrambling is shown in Fig. 2. Signal reconstructed using the scrambled signal i.e without descrambling is shown in Fig. 3. ROMP algorithm is used for reconstruction in both case for a subrate of 0.3. These plots clearly indictes that the proposed algorithm offers nearly perfect reconstruction in first case and a poor reconstruction in second case.

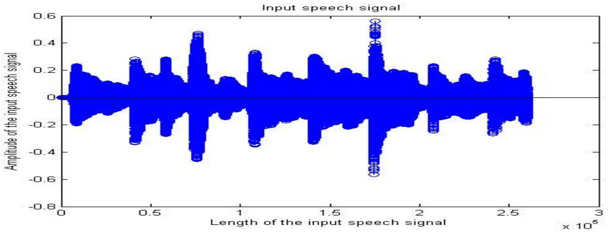


Fig. 1. Original signal

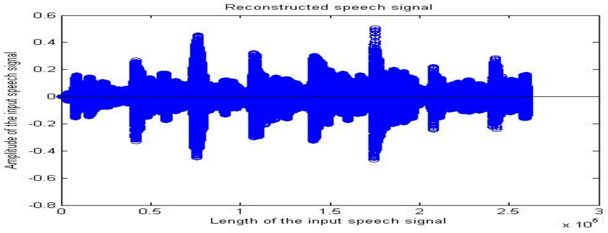


Fig. 2. Reconstructed signal after scrambling and descrambling

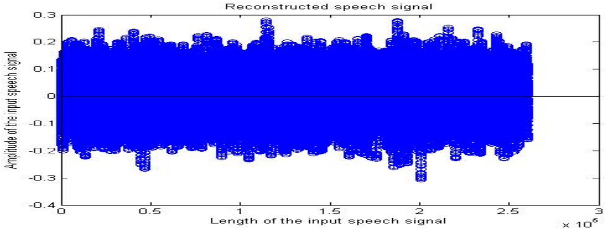


Fig. 3. Reconstructed signal after scrambling only

5.4 Resistance to Noise

Robustness in the presence of noise is tested by adding a white Gaussian noise with the scrambled audio signal, then descramble it and the resultant signal is used for reconstruction using l_1- minimisation. The noise power is varied from -60 dB to 0 dB. The results obtained for audio file *music* are shown in Fig. 4. Upto a certain value of noise power SNR remains almost constant and then it decreases rapidly. As subrate increases SNR also increases. From the graph it is evident that, the proposed method guarantees a satisfactory reconstruction performance upto a noise power of -20dB.

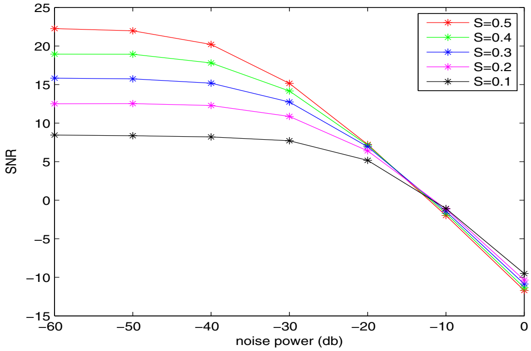


Fig. 4. SNR for different values of S for different values of noise power

6 Conclusion

A new scrambling technique for digital audio signal has been introduced. The proposed scheme takes advantage of compressive sensing and Arnold scrambling to achieve excellent compression, robustness and a high scrambling degree. The paper studies the effect of variation in subrate on SNR, scrambling degree, correlation coefficient and robustness. The method is suitable for speech and music audio files of different size. Experimental results shows that the scheme is very efficient.

References

1. Candès, E.J., Wakin, M.B.: An introduction to compressive sampling. IEEE Signal Processing Magazine 25(2), 21–30 (2008)
2. Del Re, E., Fantacci, R., Maffucci, D.: A new speech signal scrambling method for secure communications: theory, implementation, and security evaluation. IEEE Journal on Selected Areas in Communications 7(4), 474–480 (1989)
3. Donoho, D.L.: Compressed sensing. IEEE Transactions on Information Theory 52(4), 1289–1306 (2006)
4. Huang, R., Rhee, K.H., Uchida, S.: A parallel image encryption method based on compressive sensing. In: Multimedia Tools and Applications, pp. 1–23 (2012)
5. Li, H., Qin, Z., Zhang, X., Shao, L.: An n-dimensional space audio scrambling algorithm based on random matrix. Journal of Xi'an Jiaotong University 4, 005 (2010)
6. Lin, Y., Abdulla, W.H.: A secure and robust audio watermarking scheme using multiple scrambling and adaptive synchronization. In: 2007 6th International Conference on Information, Communications & Signal Processing, pp. 1–5. IEEE (2007)
7. Madain, A., Dalhoum, A.L.A., Hiary, H., Ortega, A., Alfonseca, M.: Audio scrambling technique based on cellular automata. In: Multimedia Tools and Applications, pp. 1–20 (2012)
8. Nan, L., Yanhong, S., Jiancheng, Z.: An audio scrambling method based on fibonacci transformation. J. North China Univ. Technol. 16(3), 8–11 (2004)
9. Satti, M., Kak, S.: Multilevel indexed quasigroup encryption for data and speech. IEEE Transactions on Broadcasting 55(2), 270–281 (2009)
10. Senk, V., Delic, V.D., Milosevic, V.S.: A new speech scrambling concept based on hadamard matrices. IEEE Signal Processing Letters 4(6), 161–163 (1997)
11. Servetti, A., De Martin, J.C.: Perception-based partial encryption of compressed speech. IEEE Transactions on Speech and Audio Processing 10(8), 637–643 (2002)
12. Shang, Z., Ren, H., Zhang, J.: A block location scrambling algorithm of digital image based on arnold transformation. In: The 9th International Conference for Young Computer Scientists, ICYCS 2008, pp. 2942–2947. IEEE (2008)
13. Tropp, J.A.: Greed is good: Algorithmic results for sparse approximation. IEEE Transactions on Information Theory 50(10), 2231–2242 (2004)

Cryptanalysis of Two Authentication Scheme for DRM System

Dheerendra Mishra and Sourav Mukhopadhyay

Department of Mathematics
Indian Institute of Technology Kharagpur, India
{dheerendra,sourav}@maths.iitkgp.ernet.in

Abstract. Internet based content distribution facilitates efficient platform for digital content (movies, music, text, software) trades to the remote users. It makes electronic commerce more profiting and user-friendly. However, digital content can be easily copied and redistributed over the network. At the same time, digital rights management (DRM) system emerges in the response of these drawbacks. It tries to ensure authorized content distribution so that copyright protection can be assured. Although, most of the existing DRM system supports only one way authentication, where the server verifies user's authenticity and user simply assumed that he is interacting with the correct server. It may cause server spoofing attack. In 2006, Fan et al. proposed a certificate based authentication scheme for DRM system. In 2009, Wang at al. presented a smart card based authentication scheme for DRM system using biometric keys in which user and server can mutually authenticate each other. We analyze both the schemes and show that both the schemes fail to prove their claim of resistance to most common attacks. Fan et al.'s scheme has failed to resist known session specific temporary information attack and replay attack. Moreover, it does not ensure perfect forward secrecy. Wang et al.'s scheme does not withstand insider attack and known session specific temporary information attack and have an inefficient login phase.

Keywords: Digital Rights Management, Authentication, Anonymity, Security.

1 Introduction

The advances in network technology have made internet an easy and efficient way for data transfer. The internet provides a scalable infrastructure for multimedia contents (music, movies, document, image, software, etc.) trade. It facilitates an easy access of multimedia content at low cost to the remote users. However, the content can be easily copied and redistributed over the network without degradation in the content quality. These drawbacks results rampant piracy, where piracy causes huge revenue to lose to the electronic commerce. Digital rights management (DRM) systems are developed in the response to the rapid increase in online piracy of commercially marketed multimedia products.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 184–191, 2014.

The purpose of this technology is to regulate content consumption so that unauthorized access and illegal redistribution of multimedia content can be restricted. DRM broadly refers to the set of policies, techniques and tools which manages the access control on the digital contents [1].

Most of the existing scheme for DRM systems are introduced to enhance the functionality of DRM system [2–8]. Many of the existing schemes present one way authentication in which server verifies the user's authenticity and user simply assume that he is interacting with the correct server. However, this provides the opportunity to the adversary to mislead the user by performing server impersonation attack. The schemes [5, 6] present authenticated key agreement mechanism for DRM system, where the license server and the user can authenticate each other and can establish a session key to securely transfer the digital license over the insecure public network.

In recent times, many smart card based DRM systems have been introduced to achieve portability in DRM system such that a user can play the license anytime, anywhere and on any device using the same license [9–12]. Most of the smart card based DRM systems [13, 9, 12] do not present mutual authentication mechanism and session key agreement, where the user and the server verifies the legitimacy of each other and established a session key. In 2006, Fan et al. [14] presented an authentication scheme for DRM system. We analyze Fan et al.'s scheme and find out that their scheme is vulnerable to replay attack and known session specific temporary information attack. In 2009, Wang et al. [11] presented biometric based based mutual authentication scheme for DRM system using smart card in which user and server mutual authenticate each other and established a session key. We also analyze Wang et al.'s scheme and find out that their scheme is vulnerable to insider attack and known session specific temporary information attack. Moreover, their scheme does not preserve user's anonymity and smart card cannot identify incorrect input.

The rest of the paper is organized as follows: Section 2 discusses the Fan et al.'s scheme briefly. Section 3 demonstrates the vulnerabilities of Fan et al.'s scheme. Section 4 presents the brief review of Wang et al.'s scheme. Section 5 points out the weakness of Wang'et al.'s scheme. Finally, conclusion is drawn in Section 6.

2 Review of Fan et al.'s Authentication Scheme for DRM System

In this section, we present a brief description of Fan et al.'s authentication scheme for DRM system [14] and point out some of the weaknesses of their protocol. In Fan et al.'s scheme, user and server share common information such as large prime numbers p and q, a generator g of group G_p, one way hash functions H_1, H_2, H_3, symmetric key encryption and decryption algorithms, and signature algorithm. The server also selects a secret key $X \in Z_p$ and determine public key

$Y = g^X \pmod p$. Then, the authentication protocol between user and server works as follows:

- U randomly selects $u \in Z_p$ and computes $a = g^u \mod p$, then sends $< a >$ to S.
- Upon receiving the information, S randomly selects a number $r \in Z_p$ and computes the session key $sk = H_1(a^x, r) = H_1((g^u)^x, r)$ and $b = H_2(sk, r, \mathrm{ID}_S)$, then transmits (b, r, CertS) to U.
- Upon receiving the message, U achieves y from the Cert$S$ and computes $sk = H_1(y^u, r)$, then verifies $b =? H_2(sk, r, \mathrm{ID}_S)$. If verification holds, S is identified and sk is considered session key by U. U computes $v = Sig_U(H_3(y, a, r, \mathrm{ID}_U))$, then encrypts v and his public key certificate $(CertU)$ using sk and gets $e = E_{sk}(v, CertU)$, then sends $< e >$ to S.
- Upon receiving the message, S decrypts e by sk and gets $(v, CertU)$. Then, S verifies public key certificate $CertU$. If verification holds, S verifies the signature $v = Sig_U(H_3(y, a, r, \mathrm{ID}_U))$. If verification holds, U is authorized by S.

3 Cryptanalysis of Fan et al.'s Scheme

We analyze Fan et al.'s Scheme and find out that their scheme is vulnerable to known session specific temporary information attack and replay attack. Additionally, it does not achieve forward secrecy.

3.1 Known Session Specific Temporary Information Attack

Fan et al.'s scheme does not resist known session specific temporary information attack, as if temporary secret (u) of user compromised, then an adversary (E) can compute the session key. It is clear from the following facts:

- E achieves (b, r, CertS) as E can record the transmitted messages via public channel.
- E achieves y from the Cert$S$.
- E computes session key $sk = H_1(y^u, r)$ using compromised temporary information u.

3.2 Perfect Forward Secrecy

Fan et al.'s scheme does not provides perfect forward secrecy, as the adversary can compute established session key with the compromised secret key of server. This works as follows:

- E achieves $a = g^u \pmod p$ and (b, r, CertS), as both the messages transmit via public channel.
- E computes session key $sk = H_1(a^X, r) = H_1(g^{uX}, r)$ using S's compromised secret key X.

3.3 Replay Attack

Replay attack is one kind attack in which an attack can replay the previously transmitted message from the sender to the server. An adversary can eavesdrop the communication which broadcast through public channels and can launch it easily by replaying an eavesdropped message [15, 16]. In an efficient protocol, a valid data transmission should not be maliciously or fraudulently repeated or delayed the message.

Fan et al.'s scheme fails to resist replay attack, which can be justified as follows:

- Suppose an adversary E intercepts the previous login request message and achieves $< a >$, where $a = g^u \pmod{p}$.
- E starts a new session with message $< a' > = < a >$.
- S randomly selects a number $r \in Z_p$ and computes the session key $sk = H_1(a^x, r)$

Since the server does not verify the freshness of message, therefore, server S will not be able to detect whether this message is a replayed message or not. Hence, this scheme is vulnerable to resist replay attack.

4 Review of Wang et al.'s Biometric Based Authentication Scheme for DRM System Using Smart Card

Wang et al. [11] presented a multi-modal biometrics based remote user authentication scheme for DRM system. Their scheme supports client server authentication and the server authentication. In client server authentication, it adopts watermarking technique and multi-modal biometric system. For the server authentication, their scheme uses a generalized form of the ElGamal signature scheme. Their scheme comprises the following phases:

- Registration phase
- Login phase
- Authentication phase

The brief description of Wang et al.'s scheme is as follows:

4.1 Registration Phase

The registration center RC registers the user and issues a smart card. RC computes $Y_1 = g^{X_1} \bmod p$ and $Y_2 = g^{X_2} \bmod p$, where where $1 < X_1, X_2 < p - 1$ are two private keys of servers and p is a large prime number of size 512 or 1024 bits, q is a 160 bit prime divisor of $p-1$ and g is an element of $GF(p)$ of order q. If a user U submits his registration request with identity ID_U, password PW_U, iris biometric B_1 and face biometric B_2 to registration center, then registration center performs the following steps:

- Computes $A_1 = h(\mathrm{ID}_U \oplus X_1)$, $A_2 = h(\mathrm{ID}_U \oplus X_2)$ and then $V_1 = A_1 \oplus h(PW_U \oplus B_1)$ and $V_2 = A_2 \oplus h(PW_U \oplus B_2)$.
- Personalizes U's smart card by embedding the security parameters $\{\mathrm{ID}_U, Y_1, Y_2, B_1, B_2, A_1, A_2, V_1, V_2, h(.), g, p, q\}$ and issue smart card to U.

4.2 Login Phase

When a user U wishes to login to the server, he inserts his smart card into the smart card reader and inputs ID_U, PW_U and imprints his face biometric at the sensor. If face biometric feature is successfully verified, then extract the iris feature watermark from face image. If the calculated correlation value between the extracted features of the iris and the registered features of the iris is greater than predefined threshold, the authentication succeeds and then the smart card calculates the login message as follows:

- Generate random numbers r_1 and r_2 by the minutiae extracted from the iris and the face, respectively.
- Calculate $L_i = (Y_i)^{r_i} \bmod p(i = 1, 2)$.
- Calculate $C_i = h(L_i \oplus T)$ $(j = 1, 2)$, where T is the current timestamp.
- Calculate $A_i = V_i \oplus h(PW_U \oplus S_i)$ $(i = 1, 2)$
- Compute $n_i = r_i - A_i m_i \bmod p$ $(j = 1, 2)$, where $1 \leq m_1, m_2 \leq p - 1$ then, send a login message $C_i = \{ID_U, m_i, n_i, T\}$ $(i = 1, 2)$ to the server.

4.3 Authentication Phase

Authentication phase proceeds as follows:

- Upon receiving the message $C_i, (i = 1, 2)$ at time T', S verifies the format of ID_U. If validation holds, S verifies $T' - T < \Delta T$, where ΔT denotes the valid time delay in message transmission. If verification does not hold, it declines the request. Otherwise, go to the next step.
- S computes $A_i = h(ID_U \oplus X_C)$ $(i = 1, 2)$ and $L_i = (g^{n_i} \cdot g^{A_i \cdot m_i})^{X_C} \bmod p$ $(i = 1, 2)$, then verifies $m_i =? h(L_i \oplus T)(i = 1, 2)$. If verification holds, the login request is accepted.
- S takes current timestamp T" and computes $C_i' = h(L_i, A_i, T")$ $(i = 1, 2)$, then sends the message $\{C_i', T"\}$ $(i = 1, 2)$ to U.
- Upon receiving the messages $\{C_i', T"\}$ $(i = 1, 2)$ at time T''', U verifies $T''' - T'' < \Delta T$. If verification does not hold, the session terminates. Otherwise, next step executes.
- U verifies $C_i' = h(L_i, A_i, T")$ $(i = 1, 2)$. If authentication holds, U considers the authentication of the responder.
- The value $L_i = (g^{r_i})^{X_C} \bmod p$ $(i = 1, 2)$ is used as a session key.

5 Cryptanalysis of Wang et al.'s Biometric Based Authentication Scheme for DRM System Using Smart Card

5.1 Insider Attack

In general, a user uses the same password for several accounts because it is difficult to remember several distinct passwords. When a user submits his password in its original form to the server, an insider can know the user's password. This gives the opportunities to a malicious insider to the access user's account which are protected with the same passwords. However, in Wang et al.'s scheme, the user submits his original password to the server. It makes insider attack possible.

5.2 Known Session Specific Temporary Information Attack

In this scenario, compromise of a short-term keys should not result the compromise of the session key. However, in Wang et al.'s scheme, if an adversary (E) can achieve short-term keys r_1 and r_2, then it can compute session key sk as follows:

- Achieve server public key $Y_1 = g^{X_1}$ mod$p$, as the public key is publicly available.
- Compute session key $L_i = (g^{X_C})^{r_i}$ mod p ($i = 1, 2$) using r_1 and r_2.

5.3 Inefficient Login Phase

In general, the user may keep different passwords for different accounts, services and applications to ensure security. The user may get confused some time in selection of passwords and may use password of one account for another account as human may sometimes forget the password or commit some mistake while enters a password [17].

In Wang et al.'s scheme smart card does not verifies the correctness of input in login phase. Therefore, if a user enters a wrong password $PW_U^* \neq PW_U$ even then the smart card executes the login session as follows:

- Generate random numbers r_1 and r_2 by the minutiae extracted from the iris and the face, respectively.
- Compute $L_i = (Y_i)^{r_i}$ mod $p(i = 1, 2)$.
- Compute $m_i = h(L_i \oplus T)$ ($i = 1, 2$) , where T is the current timestamp.
- Compute $A_i^* = V_i \oplus h(PW_U^* \oplus S_i)$ ($i = 1, 2$)
- Compute $n_i^* = r_i - A_i^* m_i$ mod p ($j = 1, 2$), then send a login message $C_i^* = \{ID_U, m_i, n_i^*, T\}$ ($i = 1, 2$) to the server.

The inefficiency of smart card to verify the correctness of the password, causes extra computation and communication overhead. The communication overhead is 4*128 = 512 bits to send the message $< C_i^* = \{ID_U, m_i, n_i^*, T\}$ $(i = 1, 2) >$, if $\{ID_U, m_1, m_2, T\}$ and $\{n_1, n_2\}$ are about 128 bits and 1024 bits, respectively. The computational overhead is $T_E + 2T_H + T_m + T_A + 3T_X$ where T_H, T_E, T_m, T_A and T_X denote the time complexity of hash function, exponential, multiplication, addition/substration and XOR operations, respectively.

6 Conclusion

The presented study analyzes Fan et al.'s and Wang et al.'s schemes and demonstrates their weaknesses. This investigation shows that both the schemes fail to provide an efficient authentication framework for DRM system which can resist the attacks. Moreover, the study demonstrates the flaw in Wang et al.'s scheme login phase which shows that the smart card cannot verify the correctness of input.

Acknowledgement. First author is grateful to University Grants Commission (UGC), New Delhi, India, for their financial support.

References

1. Ku, W., Chi, C.: Survey on the technological aspects of digital rights management. Information Security, 391–403 (2004)
2. Dutta, R., Mishra, D., Mukhopadhyay, S.: Vector space access structure and ID based distributed DRM key management. In: Abraham, A., Mauri, J.L., Buford, J.F., Suzuki, J., Thampi, S.M. (eds.) ACC 2011, Part IV. CCIS, vol. 193, pp. 223–232. Springer, Heidelberg (2011)
3. Liu, Q., Safavi-Naini, R., Sheppard, N.P.: Digital rights management for content distribution. In: Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003, vol. 21, pp. 49–58. Australian Computer Society, Inc. (2003)
4. Michiels, S., Verslype, K., Joosen, W., De Decker, B.: Towards a software architecture for DRM. In: Proceedings of the 5th ACM Workshop on Digital Rights Management, pp. 65–74. ACM (2005)
5. Mishra, D., Mukhopadhyay, S.: A certificateless authenticated key agreement protocol for digital rights management system. In: Singh, K., Awasthi, A.K. (eds.) QShine 2013. LNICST, vol. 115, pp. 568–577. Springer, Heidelberg (2013)
6. Mishra, D., Mukhopadhyay, S.: Secure content delivery in DRM system with consumer privacy. In: Deng, R.H., Feng, T. (eds.) ISPEC 2013. LNCS, vol. 7863, pp. 321–335. Springer, Heidelberg (2013)
7. Nair, S.K., Popescu, B.C., Gamage, C., Crispo, B., Tanenbaum, A.S.: Enabling drm-preserving digital content redistribution. In: Seventh IEEE International Conference on E-Commerce Technology, CEC 2005, pp. 151–158. IEEE (2005)
8. Nützel, J., Beyer, A.: Towards trust in digital rights management systems. In: Fischer-Hübner, S., Furnell, S., Lambrinoudakis, C. (eds.) TrustBus 2006. LNCS, vol. 4083, pp. 162–171. Springer, Heidelberg (2006)
9. Sun, H.M., Hung, C.F., Chen, C.M.: An improved digital rights management system based on smart cards. In: Digital EcoSystems and Technologies Conference, DEST 2007, pp. 308–313. Inaugural IEEE-IES, IEEE (2007)
10. Fourar-Laidi, H.: A smart card based framework for securing e-business transactions in distributed systems. Journal of King Saud University-Computer and Information Sciences 25(1), 1–5 (2013)
11. Wang, D., Li, J., Memik, G.: Authentication scheme of DRM system for remote users based on multimodal biometrics, watermarking and smart cards. In: WRI Global Congress on Intelligent Systems, GCIS, vol. 2., 530–534. IEEE (2009)
12. Lee, N.Y., Lee, T.Y.: User friendly digital rights management system based on smart cards. In: Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2009, pp. 869–872. IEEE (2009)

13. Jeong, E.S., Sur, C., Rhee, K.H.: A new DRM system based on graded contents sharing and time-block distribution for home networks. In: 6th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2007, pp. 830–833. IEEE (2007)
14. Fan, K., Pei, Q., Mo, W., Zhao, X., Li, X.: A novel authentication mechanism for improving the creditability of drm system. In: International Conference on Communication Technology, ICCT 2006, pp. 1–4. IEEE (2006)
15. Malladi, S., Heckendorn, A.F.J.,, R.B.: On preventing replay attacks on security protocols. Technical report, DTIC Document (2002)
16. Aura, T.: Strategies against replay attacks. In: Proceedings of 10th Computer Security Foundations Workshop, pp. 59–68 (1997)
17. Mishra, D.: A study on id-based authentication schemes for telecare medical information system. arXiv preprint arXiv:1311.0151 (2013)

Improving Test Conformance of Smart Cards versus EMV-Specification by Using on the Fly Temporal Property Verification

Germain Jolly, Sylvain Vernois, and Jean-Luc Lambert

Normandie Univ., UNICAEN, GREYC, ENSICAEN, CNRS
6, Boulevard Marchal Juin - F-14050 CAEN Cedex - France
{germain.jolly,sylvain.vernois}@ensicaen.fr,
jean-luc.lambert@unicaen.fr

Abstract. Electronic payment transactions using smart card are based on the Europay Mastercard Visa (EMV) specifications. This standard appeared in 1995 in order to ensure security and global interoperability between EMV-compliant smart cards and EMV-compliant payment terminals throughout the world. Another purpose of EMV specifications is to permit a secure control of offline credit card transaction approvals. This paper will expose a way to improve verification and validation of the payment application stored in the chip of the smart card based on temporal property verification. In fact, each issuer (e.g., MasterCard) defines its own EMV-compliant specification, allowing different implementation cases and possible errors and we discuss about a method to detect anomalies to avoid smart card vulnerabilities. The properties will be designed in conformance with EMV-specification but our goal is not to formally prove them. We consider implementations through a black-box testing approach, therefore we cannot prove the properties as we don't have access to the source code. However, we can observe the command/response exchanges and detect, on the fly, when an expected property is violated.

Keywords: Payment, EMV, Smart Card, Evaluation, Temporal Property.

1 Introduction

According to EMVco, there were 1.55 billion EMV [1] compliant chip-based payment cards in use worldwide in 2012 and it's still growing. The mass deployment amplifies the security risks and need for the manufacturers to have a head start over the capabilities of attackers. Once created and before being sold, the cards must be certified by a certification authority. This authority will appoint centers specialized in the analysis of vulnerabilities to check if the security schemes are respected. The cards must be verified in order to be validated to be used in everyday life. Regarding the validation of payment applications (e.g., MasterCard applications [2]), we can generate automatically a large number of test cases, for example with fuzzing technics [3]. The problem is that it is difficult

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 192–201, 2014.

for a campaign of intensive testing to trace the root reason of a malfunction as steps triggering the error can be generated before the detection of the malfunction. We propose to detect the violation of properties. Properties will be defined in order to understand the cause of a malfunction during the transaction between the terminal and the smart card.

First, the background and limits of evaluation methods will be studied. The second part deals with the main idea and the architecture of a tool created for this study. In the third part, we will talk about our contribution on verification and validation methods with property definition. Finally, the tool realized using the framework WSCT [4] will be exposed.

2 Background

2.1 Communication between the Terminal and the Smart Card

An electronic transaction may be divided into several transactions which are each a coherent set of information exchanged through a network. Transfered data from the terminal to the smart card is called Command APDU (Application Protocol Data Unit) and received data by the terminal from the smart card is called Response APDU. For each command emitted by the terminal, the card sends back one response and acts as a slave in the dialog. It is defined in ISO/IEC 7816 standard [5] and exposed on the figure 1.

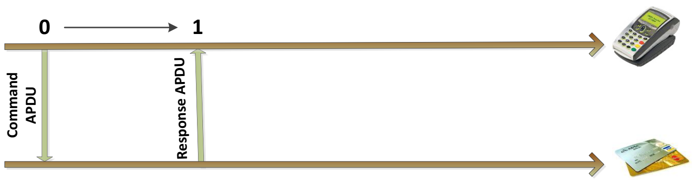


Fig. 1. A pair command/response APDU

On the figure 2, we can see the structure of a Command APDU and a Response APDU according to ISO/IEC 7816. CLA indicates the type of the command, INS the specific command, P1 and P2 are parameters for the command, LC indicates the length of the UDC which is optional data and LE indicates the length of the expected data. This expected data, only contained in the response if LE is in the command, is the UDR. Finally, SW1 and SW2 are mandatory in the response and are the command processing status.

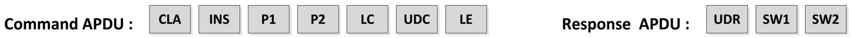


Fig. 2. Composition of a Command APDU and a Response APDU

2.2 Card Specifications: M/CHIP

In our case, the MasterCard M/CHIP specification [2], an EMV-compliant specification for Mastercard smart cards, is studied. The application contained on the chip can take several states (idle, selected, ...). The evolution of the state of the application is allowed by sending Command APDU (select, get data, GPO, ...) and receiving Response APDU (9000, 6283, ...). A machine state is given by Mastercard to illustrate the M/CHIP application. The possible application states are:

- idle : Application is not currently selected
- selected : Application is selected
- initiated : Transaction is initiated
- online : Application expects a connection with the issuer
- script : Application is ready to accepta script command

The payment application can evoluate only by receiving and responding to a serie of pairs command/response, this is the concept of application's state. We can't only consider the acceptation of a single command but the response taking into account its current state and then its past evolution.

2.3 Limits of Evaluation Methods

The smart cards must be certified before being issued by a financial institution. Many evaluation methods exist to help verify and validate smart cards, e.g., fuzzing allows to generate automatically a large number of test cases [3]. However, these methods show a lack of visibility. We can only know if a smart card is correct. We would like to know where is the error of implementation. We need to know how to detect and repair an error with more visibility. Others methods are knowns on Java Card applets like the use of pre- and postconditions is seen in [6] or [7]. But this method is a white box method. In our case, we would like to study marketed smart cards in addition of modified smart cards. We consider payment application, i.e., java card applets through a black box testing approach.

3 A Tool to Observe the Payment Transaction

To obtain more information about an error, we can use a temporal analysis, i.e., check that sequences of Commands/Responses APDU during the transaction are correct. This method is often used in the field of electronics. Assertion Based Design [8] allows to verify properties, relationships or sequences on electronics systems. A clock signal is used to know when the system is evolving and when the properties must be verified. In this context, we will do a temporal analysis to improve the evaluation of a smart card.

3.1 WSCT Framework

WSCT [9] is a framework developped by S. Vernois *et al.* for several years. It is written in C# and allows to work with smart cards, e.g. for exploration and finding fault on smart cards [4]. The two main purposes of this tool are to provide :

- an API object-oriented to access a smart card reader.
- an evolutive GUI with creation of plugins to manipulate smart cards.

3.2 Temporal Analysis

The main purpose is to improve the known evaluation method by creating an independant module able to observe the system behaviour and to detect when a property is violated. As the state of the application can change only by sending and receiving command/reponse APDU, we can detect improper behavior by observating the input and output of the smart card. These two events are legitimate candidates to define the clock timer that will be used to launch the verification of properties, similarly to a clock signal in the field of semi-conductor with the Assertion Based Design [8].

Using this clock, we can define assertions, properties, sequences on systems. The signals A, B and C can be associated to define a sequence, i.e., $S : A \; and \; B \; and \; \overline{C}$. This sequence will be true only on the second clock cycle. On the figure 3, you can see the illustration of a clock and its association safety property and signals.

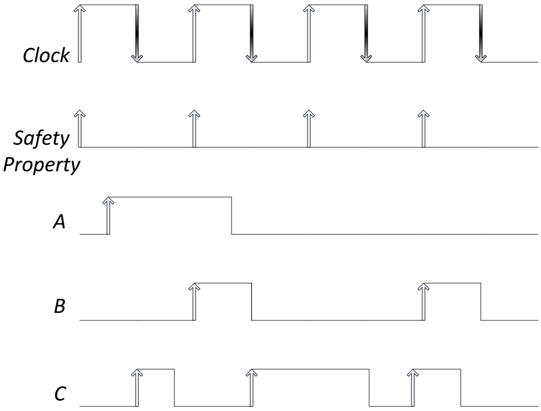


Fig. 3. Clock defined by Command APDU and Response APDU

We observe the Commands and Responses between the chip and the terminal. Then we may observe a property by seeing Commands and Responses like signals. Accessing data that normally doesn't exist on the card application would be a simple property. An other property, a temporal one, could check the refusal of a command, e.g., a *Get Processing Option* command that initiates the transaction within the card after acceptance of a command that would have normally put the application in a state normally able to accept it. Such error of implementation has already been discovered and are difficult to diagnose.

3.3 Objective

We will simulate the transaction, which can possibly be a fuzzing transaction or a nominal transaction, using the WSCT framework. Previous studies have been done on fuzzing using WSCT. We could use this work and add our tool to improve the results of this work by a higher level of visibility. The goal is not to replace existing work on validation but to improve it. The smart card is connected to WSCT, which can act as a terminal sending preprocessed commands to the card. The observation tool, which purpose is to do the temporal analysis previously presented on smart card applications, is a WSCT plugin made of three modules: an observer to examine the sending and reception of commands and responses, a detector to identify the sequences and a property detector in order to inform the customer, terminal or other entity when a sequence is seen and then a property is violated. These sequences must satisfy some predefined properties and each violated property means there is an incorrect behavior. Figure 4 illustrates how the tool is linked to the transmitted data during the transaction.

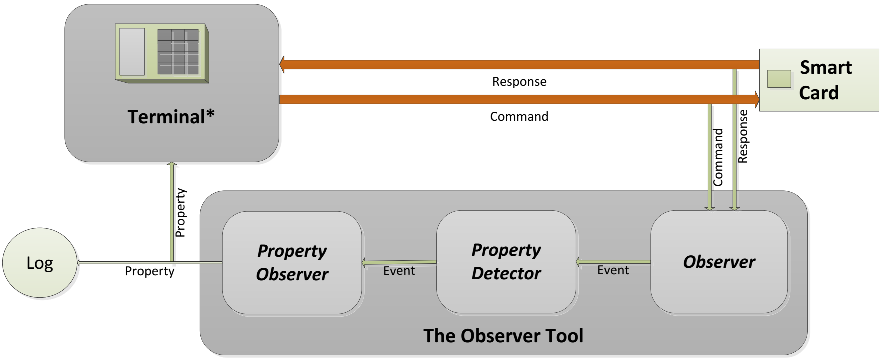


Fig. 4. Diagram of the observer and the associated interactions

The main purpose is to be sure of the validation of a payment application by detecting errors of implementation. We must see it as an additionnal level of validation to improve the confidence of the smart card. We want to emphasize the independence of the tool in comparison with communication between the terminal and the card but also the modularity concerning the properties. Several libraries of properties can be created for different smart card applications.

And we could easily add or delete properties for the current analysis in order to verify specific part of the implementation or a specific application.

4 Contribution

4.1 Property Definition

An improved clock can be defined on the figure 5. By fixing an origin point (0), we can study evolution of the smart card, more precisely the payment application, by studing series of n pairs of command/response. Finally, a property is a local theoretical evolution of the payment application. We can define two kinds of properties : simple properties (only on one pair command/response) and temporal properties (defined with several pairs). The commands match the rising clock cycles. The properties are designed in conformance with EMV-specification. The only thing we can do is to observe the command/response exchanges and detect, on the fly, when an expected property is violated. Indeed, we are not suppose to know the source code.

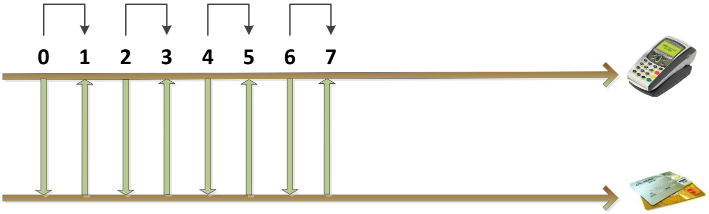


Fig. 5. The clock defined during APDU communication

Technically, a property defines a local behavior, i.e., that under certain predicates at a given moment, the smart card must have a specific behavior. As properties are defined as associations of commands and responses, we can describe all kind of features because we can observe the behavior of the application by observing the APDU communication. The commands are associated with even numbers and the responses with odd numbers. Therefore, we can only define properties with CLA, INS, P1, P2, LC, UDC and LE on even clock cycles and with UDR, SW1 and SW2 on odd clock cycles. The value of some fields involve a reaction from the smart card and consequently the value of others fields. The property must be checked to ensure that the local behavior of the smart card is correct. P_0 shows the most natural structure of a property, as an implication. By logical equivalence, we can use a second form to define P_0. Indeed, we have *A implies B* equivalent to *B or Not(A).*

$$P_0 \ : \ (field(0) \ and \ ... \ and \ field(i)) \ \Rightarrow \ (field(1) \ and \ ... \ and \ field(j))$$

$$P_0 : (field(1) \ and \ ... \ and \ field(j)) \ or \ \overline{(field(0) \ and \ ... \ and \ field(i)})$$

$$with \ \ 0 < i < j$$

4.2 Simple Properties and Predicates on One Time Clock Only

We expose two simple properties in order to illustrate our work. Indeed, a simple thing is to verify the integrity of the command and response. The property P_1 check the correctness of the response. Is the sent data by the smart card correct according the received command? Here, we are verifying that the length of the expected data (UDR) in the response is LE, given in the previous command. In this case, only one time clock is checked (we are studying the 0 and 1 time clock).

Response length correct :

$$P_1 \ : \ ((UDR(1) \ \neq \ 0) \ and \ (UDR(1).length \ = \ LE(0)) \ or \ \overline{(LE(0) \neq 0)}$$

This figure 1 illustrates a simple property to show the link between the sent and received data and the property definition. It's the generalization of the two previous properties. The command (defined by an association of fields with specific values) implies the response (defined by an associated of fields with specific values). We can also verify the value of the fields contained in the command plus the value of the fields contained in the response.

4.3 Complexes Properties Defined on n Times Clock

We can check series of pairs command/response APDU too. According to one send command, the property P_2 allows to detect a replay. In fact, if every bytes of the second command are the same than the first command, this property will be false. For each two command/response pairs, we have to check the property is correct (at least one of the fields of the second command is different) to avoid replay during the transaction.

No command replay detected :

$$\begin{aligned} P_2 : & (CLA(2) \neq CLA(0)) \ or \ (INS(2) \neq INS(0)) \ or \ (P1(2) \neq P1(0)) \ or \\ & (P2(2) \neq P2(0)) \ or \ (LC(2) \neq LC(0)) \ or \ (UDC(2) \neq UDC(0)) \ or \\ & (LE(2) \neq LE(0)) \end{aligned}$$

The last exposed property P_3 permit to detect an error of implementation in the application. We can see the theorical behaviour of the smart card notably between the "selected" state and the "initiated" state using only the GPO command ('A8' is the INS for Get Processing Options Command) in the MasterCard specification [2]. We could generalize it to detect any wrong behavior between these two states.

Good behavior with three GPO commands :

$$P_3 : \frac{((SW1(1) = 90)\ and\ (SW2(1) = 00)\ and\ (SW1(3) \neq 90)\ and\ (SW2(3) \neq 00)\ and\ (SW1(5) = 90)\ and\ (SW2(5) = 00))\ or}{(INS(0) = A8)\ and\ (INS(2) = A8)\ and\ (INS(4) = A8)}$$

The figure 5 illustrates the P_3 property that involves several pairs command/response, so three clock cycles. The commands associated to the number 0, 2 and 4 of the clock timer are defined by an association of fields with specific values and implies the responses associated to the number 1, 3 and 5 defined by an associated of fields with specific values. In fact, the last response is due to the previous commands. The property is finally a set of values to be checked and is true if the application is correct.

5 The Observation Tool

5.1 WSCT and the Observation Plugin

In order to do the tool presented in the figure 4, we have created two plugins on WSCT. We have grouped the three modules : the Observer, the Property Detector and the Property Observer into one plugin. In fact, its purpose is to take as inputs the APDU commands and responses and compute if the properties are checked or not, in order to verify the smart card with a higher level of appreciation. The plugin called Transaction do the communication with the smart card. On the figure 6, you can see how we have grouped the modules and linked them with WSCT. Two plugins have been created : one, called *Transaction*, allows to launch any type of terminal (fuzzing, nominal, ...) [10] and the other, called *Observer*, is the tool to observe properties.

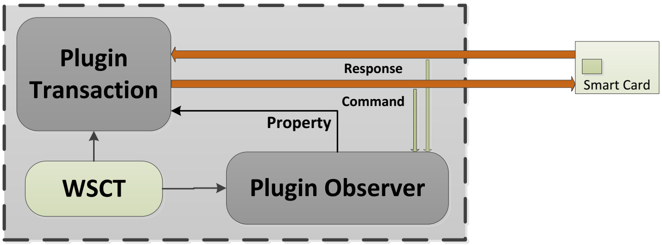


Fig. 6. WSCT, its perimeter and the first version of the tool

From APDU communication, the Observer plugin can construct all the properties to be verified. If one property is violated, the user must be warned. We can observe independantly of the transaction processing the data exchanged between a smart card and the transaction application. Violation of simple properties can also be detected, e.g., detecting wrong command or replay.

5.2 Verification of Properties Using This Tool

The figure 7 describes the algorithm to verify properties. One property is an object using variables containing the value of commands and responses APDU captured at a given moment on a discrete and bounded time-line (0, 1, 2...n). We instantiate the variable of the properties by capturing APDU communication on the fly. When all the variables of a property have been instantiated, we can verify their validity and interrupt the terminal or send a signal to the user if there is a violation. Then all the properties which verification is done (all their varibales were instantiated) are killed and new ones are created that will check the same properties in the forthcoming instants.

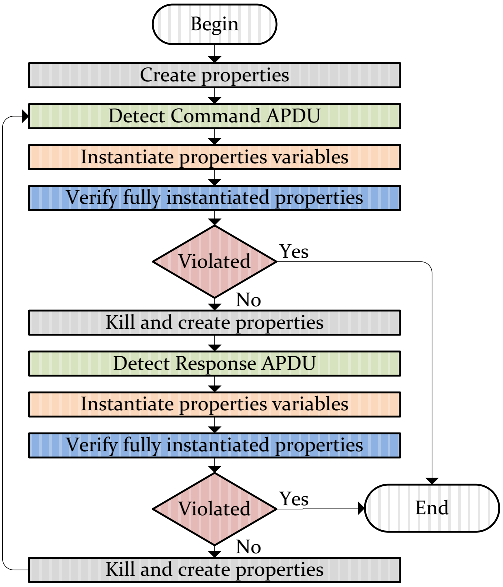

Begin
Create properties
Detect Command APDU
Instantiate properties variables
Verify fully instantiated properties
Violated
Yes
No
Kill and create properties
Detect Response APDU
Instantiate properties variables
Verify fully instantiated properties
Violated
Yes
End
No
Kill and create properties

Fig. 7. Verification process of properties

5.3 Advantages of Our Verification Method

Finally, this method shows four main advantages :

- Black-box verification : We need neither the source code of the smart card neither the source code of the terminal;
- Generic verification : This method can be used on other kinds of smart card application;

– On the fly verification : This method is independant of the transaction process by only capturing the communication;
– Modular verification : A library of properties or just a specific property can be verified.

6 Conclusion

In this paper, we have exposed an additional method to improve software evaluation of smart cards. We can easily isolate a property on the fly to obtain more information about a detected error. This study, called temporal analysis, allows us to know what really happened and especially when it happened through the use of temporal properties. Through a black box method, we are able to determine sequences of data exchanged between the smart card and the terminal which permit us to detect wrong behavior of the payment application. Finally, we can strengthen the validation of smart cards by using our method over the methods already used.

This work will be continued and we will focus on several main topics. First, the use of temporal properties for software evaluation must be validated. To evaluate our work, we will use validated smart cards and modified smart cards containing errors. This method will be used on several types of applications, including M/CHIP applications. The next step of this work is an automatic generation of properties to verify on a specific application. The tool should allow to learn properties all along the execution too in order to become a flexible and evolving tool.

References

1. EMV Integrated Circuit Card Specifications for Payment Systems, version 4.3 EMVco (2011)
2. M/Chip 4 Card Application Specifications for Credit and Debit, MasterCard International (2002)
3. Lancia, J.: Un framework de fuzzing pour cartes a puce: application aux protocoles EMV (2011)
4. Vernois, S., Alimi, V.: WinSCard Tools: a software for the development and security analysis of transactions with smartcards (2010)
5. ISO/IEC 7816, International Organization for Standardization and the International Electrotechnical Commission
6. Philippaerts, P., Mhlberg, J.T., Penninckx, W., Smans, J., Jacobs, B., Piessens, F.: Software Verification with Verifast: industrial Case Studies (2013)
7. Distefano, D., Parkinson, M.J.: Jstar: Towards Practical Verification for Java (2009)
8. Foster, H.D., Krolnik, A.C., Lacey, D.J.: Assertion-Based Design (2010)
9. Source code of WSCT, https://github.com/wsct
10. Vibert, B., Alimi, V., Vernois, S.: Analyse de la sécurité de transactions à puce avec le framework WinSCard Tools (2012)

Palmprint Recognition Using Fusion of 2D-Gabor and 2D Log-Gabor Features

Munaga V.N.K. Prasad, Ilaiah Kavati, and B. Adinarayana

Institute for Development and Research in Banking Technology (IDRBT),
Castle Hills, Masab Tank, Hyderabad-57, India
mvnkprasad@idrbt.ac.in,
{kavati089,contact.adinarayana}@gmail.com

Abstract. Palmprint technology is a new branch of biometrics used to identify an individual. Palmprint has rich set of features like palm lines, wrinkles, minutiae points, texture, ridges etc. Several line and texture extraction techniques for palmprint have been extensively studied. This paper presents an intra-modal authentication system based on texture information extracted from the palmprint using the 2D- Gabor and 2D-Log Gabor filters. An individual feature vector is computed for a palmprint using the extracted texture information of each filter type. Performance of the system using two feature types is evaluated individually. Finally, we combine the two feature types using feature level fusion to develop an intra-modal palmprint recognition system. The experiments are evaluated on a standard benchmark database (PolyU Database), and the results shows that significant improvement in terms of recognition accuracy and error rates with the proposed intra-modal recognition system compared to individual representations.

Keywords: Palmprint, Gabor filter, Log-Gabor filter, Intra-modal, Feature Level Fusion.

1 Introduction

Biometrics plays a vital role in today Networked Security world because of its reliability and uniqueness. A biometric system is basically an automated pattern recognition system that either makes identification or verifies an identity by establishing the probability that a specific physiological or behavioral characteristic is valid [7]. Various biometric technologies were developed during the past few decades such as fingerprint, palmprint, iris, face, voice, signature and hand for real-time identification. However, compared with other biometrics, Palmprint has distinct advantages like larger palm area, stable line features, noncontact, low resolution image, low cost acquisition device and more user acceptance etc. Further, the palmprint contain rich set of features like thick principal lines[1-2], ridges and wrinkles which are stable throughout the life time. For these reasons, the palmprint recognition has attracted an increasing attention from researchers during the past few years, and it has been proven that palm based recognition is one of the best among the others [3].

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 202–210, 2014.

The palmprint based recognition systems can be broadly categorized into three categories: 1) line based [12,13], 2) appearance based [14-16], and 3) texture base approaches [17-25]. In [12], authors used Sobel and morphological operations to extract line features from palmprints. Ridges of the palmprint by eliminating creases has been proposed in [13]. Systems based on appearance features of palmprint include Principle Component Analysis [14], Fisher Discriminate Analysis [15], Independent Component Analysis [16]. In addition to this, the researchers also employ texture based approaches like Wavelets [17], Discrete Cosine Transform [18], Fourier Transforms [19], Scale Invariant Feature Transform [20], Speeded-up Robust Features [21], Gabor Filter [22] and its variants Competitive Code [23], etc. In this paper, we use 2D Gabor and 2DLog- Gabor filters to develop an intra-model palmprint recognition system.

The rest of the paper is organized as follows. Section 2 explains the texture extraction process using 2D Gabor filter and 2D Log-Gabor filter. Further, the computation of a Feature Vector (FV) from the extracted texture and matching of the palmprints using this feature vectors discussed in section 3. Section 4 describes the proposed fusion technique. The experiments and results from this work are presented in Section 5. Conclusions are described in section 6.

2 Texture Extraction

Gabor filters has been extensively studied in the literature to extract texture features from biometrics like fingerprint [4], Palmprint recognition [5-6], etc. The advantage with Gabor filters for palmprint is that, the extracted texture information includes principal lines, wrinkles, ridges, etc.

2.1 2D Gabor Filter

The 2D Gabor filter is a composite function with two components: a Gaussian shaped function and a complex plane wave [24]. It has the following form,

$$G(x, y, \theta, u, \sigma) = \left(\frac{1}{2\pi\sigma^2}\right) * \exp\left\{-\frac{x^2 + y^2}{2\sigma^2}\right\} * \exp\left\{2 * \pi * i * u(x\cos\theta + y\sin\theta)\right\} \quad (1)$$

where x, y represents the coordinates of the filter, ‘u’ denotes the filter center frequency, ‘σ’ is the width of Gaussian envelope, ‘θ’ is the orientation of the filter and *i* = $\sqrt{-1}$. In the experiment, we chosen the optimized values for Gabor filter parameters empirically and they are: u=0.096, σ=7.1 and θ = 45^0 .

According to Euler formula, Gabor filter can be decomposed into two parts: real part and imaginary part. The response of a Gabor filter to an image is obtained by a 2D convolution. Let *I*(*x*,*y*) donate an image and *I'*(*x*,*y*) denotes the response of the Gabor filter. The Gabor response to an image is defined as follows:

$$I'(x, y) = I(x, y) \otimes G(x, y) \quad (2)$$

The Gabor filtered image has both real and imaginary components. The magnitude of the Gabor filtered image is calculated using,

$$|I'(x,y)| = \sqrt{ReI'(x,y)^2 + ImI'(x,y)^2} \tag{3}$$

where *ReI'*(*x,y*) and *ImI'*(*x,y*) are the real and imaginary parts of the Gabor filtered image, respectively. The texture information of a palmprint after convolving with the 2D Gabor filter is shown in Fig. 1(b).

2.2 2D Log-Gabor Filter

2D Log-Gabor filter has been extensively studied to extract the texture features [8,9]. It has the following form,

$$\mathrm{G(w,v)=exp}\left\{\frac{-[\lg(w/w_0]^2}{2[\lg(k)]^2}\right\}\mathrm{exp}\left\{\frac{-[\lg(v/v_0)]^2}{2[\lg(l)]^2}\right\} \tag{4}$$

Where w_0 and v_0 represents the 2D filter center frequencies in vertical and horizontal directions respectively, and *k*, *l* are a chosen constant to control the filter bandwidth. The inverse Fourier transform of 2D Log-Gabor function is represented as,

$$\mathrm{g(x,y)} = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\mathrm{G(w,v)\, e^{-jwx}\, e^{-jvy}\, dwdv} \tag{5}$$

The response of a Log- Gabor filter to an image is obtained by a 2D convolution. Let *I*(*x,y*) donate an image and *I'*(*x,y*) denotes the response of the Log- Gabor filter. The Log- Gabor response to an image is defined as follows:

$$I'(x,y) = I(x,y) \otimes g(x,y) \tag{6}$$

The Log- Gabor filtered image has both real and imaginary components as the response of the filter is complex. The magnitude of the Log- Gabor filtered image is calculated using,

$$|I'(x,y)| = \sqrt{ReI'(x,y)^2 + ImI'(x,y)^2} \tag{7}$$

where *ReI'*(*x,y*) and *ImI'*(*x,y*) are the real and imaginary parts of the Log-Gabor filtered image, respectively. The texture information of a palmprint after convolving with the 2D Log-Gabor filter is shown in Fig. 1(c).

3 Computation of Feature Vector and Matching

This section explains the proposed method of computing feature vector from the extracted texture information for a palmprint image and the matching algorithm.

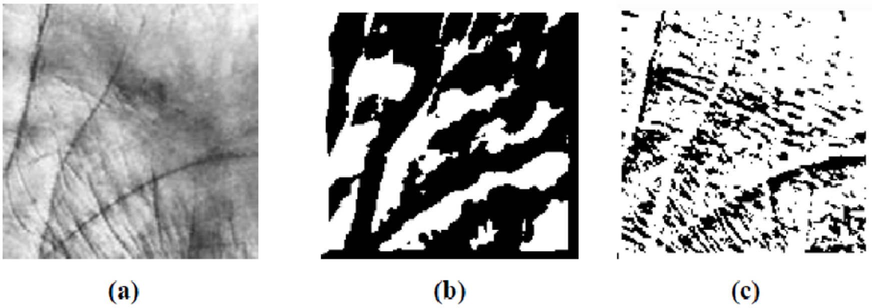


Fig. 1. (a) Original Palmprint image (b) image after convolving with 2D Gabor filter (c) images after convolving with 2D Log-Gabor filter

3.1 Feature Vector

After extracting the texture information using a particular filter, a feature vector (or template) is computed for each palmprint image. The convolved palmprint image is segmented into *n* non-overlapping sub images of equal size. Then for each sub image, standard deviation (*SD*) of its coefficients is calculated. The *SD*'s of all sub images are arranged in raster scan order to generate the feature vector (FV).

$$FV = \{SD_1, SD_2, \ldots\ldots.SD_n\} \tag{8}$$

3.2 Matching

Many existing approaches including neural networks, Hidden Markov models and correlation filters have been examined various measures for matching including cosine measure, weighted Euclidean distance, Euclidean distance and hamming distance, Pearson correlation coefficient, etc. [10]. The linear or Pearson correlation coefficient is the most widely used measurement of association between two vectors. The similarity between two templates can be measured using their correlation. Templates belonging to the same identity are expected to have a strong positive correlation. Templates belonging to different identities are expected to be uncorrelated. In this approach, we use Pearson correlation coefficient to calculate the similarity between two images.

Let $\boldsymbol{X} = \{(x_i) \mid 1 \le i \le n\}$, $Y = \{(y_i) \mid 1 \le i \le n\}$ be the two templates for which we want to calculate the degree of association. The linear correlation coefficient $r(X,Y)$ is given by the formula:

$$r(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{9}$$

where -1 ≤ r ≤ +1, and $\bar{x}, \bar{y}$ represents the mean of templates X,Y respectively. If the value of r = +1, it indicates a strong correlation and the two templates are identical, whereas a r = -1 indicates weak correlation and the templates are perfect opposite.

4 Fusion of Feature Vectors

The objective of the data fusion is to improve the performance of the system. The uni-model biometric systems may suffer due to issues such as limited population coverage, less accuracy, matcher limitations, noisy data etc [25]. Hence, uni-model biometrics may not be reliable and to overcome these limitations and improve the performance fusion of multiple biometric information has been proposed. The multiple pieces of biometric evidences can be combined by four fusion strategies: (a) fusion at the sensor level (b) fusion at the feature extraction level, (c) fusion at the matching score level, (d) fusion at the decision level. In this paper, we use the feature level fusion that combines the feature vectors of a palmprint which are obtained using the 2D Gabor filter as well as 2D Log-Gabor filter.

4.1 Feature Level Fusion

The data obtained from each feature extraction method is used to compute a new feature vector. As the texture features extracted from one method are independent of those extracted from the other, it is reasonable to concatenate the two vectors into a single new vector.

$$\mathrm{FV}_{(i,\mathrm{F})} = \{FV_{(i,G)}, FV_{(i,L)}\} \tag{10}$$

Where $FV_{(i,G)}$ is a feature vector obtained after applying Gabor filter on the palmprint image i, $FV_{(i,L)}$ is a feature vector obtained after applying Log-Gabor filter, and $FV_{(i,F)}$ is resultant feature vector after fusion. Pictorial representation of the proposed fusion approach can be observed from Figure 2.

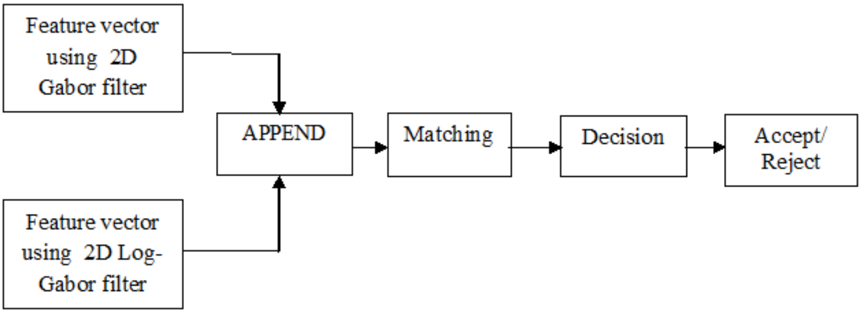
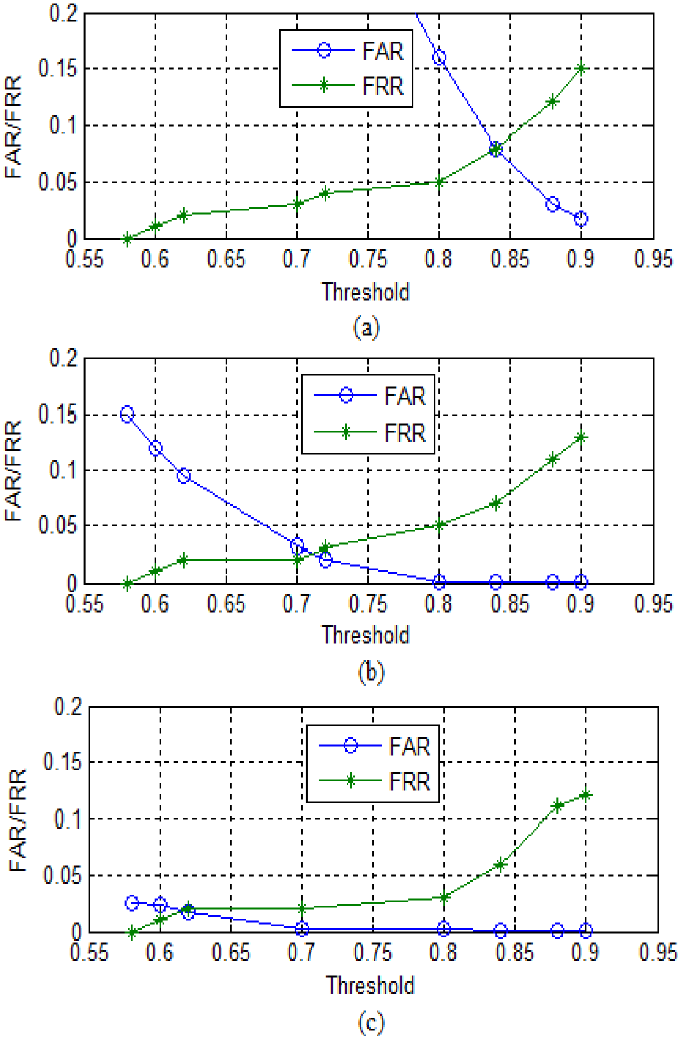

Feature vector using 2D Gabor filter
Feature vector using 2D Log-Gabor filter
APPEND
Matching
Decision
Accept/ Reject

Fig. 2. Block diagram of proposed fusion scheme

5 Experimental Results

We experimented our approach on benchmark PolyU Palmprint database [11]. PolyU database consists of 7,752 grayscale palm print images from 193 users corresponding to 386 different palms. The palm images are cropped to 150×150 pixels. The performance of the proposed authentication system is determined using four measures, namely: 1. False Acceptance Rate (FAR), 2. False Rejection Rate (FRR) 3. Genuine Acceptance Rate (GAR) and 3. Equal Error Rate (EER). FAR is the frequency that a non authorized person is accepted as authorized while FRR is the frequency that an



(a)

(b)

(c)

Fig. 3. FAR and FRR at various thresholds for different techniques: a) Gabor features based, b) Log-Gabor features based, and c) Proposed fusion technique

authorized person is rejected access and GAR is the frequency that a authorized person is accepted as authorized. The EER refers to a point where the FAR equals the FRR; a lower EER value indicates better performance.

Experiments have been carried out to evaluate the performances of the proposed fusion method, algorithm based on 2D Gabor features and algorithm based on 2D Log-Gabor features. All the techniques have been implemented using MATLAB. The performance curves plotting the FAR and FRR at various thresholds for different techniques are shown in Fig. 3. It is observed that FAR reduces as threshold increases and FRR increases with threshold.

It is observed that, the EER of the systems using Gabor and Log-Gabor features is 0.08 and 0.03 respectively. From the result, it is evident that the Log-Gabor features performs well compared to Gabor features.

The performance of proposed fusion technique can also be seen from the Fig. 6. The EER of the proposed fusion method is only 0.02 which is less than both the Gabor and Log-Gabor features methods. Table 1 shows the EER values of different methods.

In order to visually describe the performance of a biometric system, Receiver Operating Characteristics (ROC) curves are usually used. A ROC curve shows how the FAR values are changed relatively to the values of the GAR and vice-versa. The ROC curve for the proposed methods is given in Fig. 4. It is observed that, the proposed fusion system performs well as GAR is very high and FAR is very low compared to Gabor and Log- Gabor based methods.

Table 1. Performance of the different systems on PolyU database

Method	**EER**
Gabor features	0.08
Log-Gabor features	0.03
Proposed fusion method	0.02

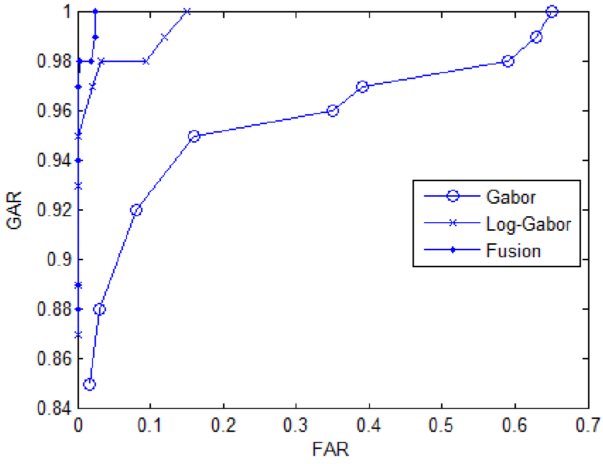


Fig. 4. ROC curves for different techniques

6 Conclusions

This paper proposes an efficient intra-model palmprint authentication system. The technique uses the Gabor and Log-Gabor features efficiently to make the system more robust. The individual performance of each feature is evaluated and it is observed that the Log-Gabor features are performed well compared to Gabor features of a palmprint. Finally, the Gabor and Log-Gabor feature vectors are fused using feature level fusion, and further improves the performance of the authentication system.

References

1. Wu, X., Zhang, D., Wang, K.: Palm Line Extraction and Matching for Personal authentication. IEEE Transactions on Systems Man and Cybernetics Part A: Systems and Humans 36, 978–987 (2006)
2. Kumar, A., Wong, D.C.M., Shen, H.C., Jain, A.K.: Personal Verification using Palmprint and Hand Geometry Biometric. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, pp. 668–678. Springer, Heidelberg (2003)
3. Han, C.C., Cheng, H.L., Lin, C.L.: Personal authentication using palm print features. In: 5th Asian Conference on Computer Vision (ACCV), pp. 23–25 (2002)
4. Chin, Y.J., Ong, T.S., Goh, M.K.O., Hiew, B.Y.: Integrating Palmprint and Fingerprint for Identity Verification. In: Third International Conference on Network and System Security, pp. 437–442 (2009)
5. Zheng, P., Sang, N.: Using Phase and Directional Line Features for Efficient Palmprint Authentication. In: 2nd International Congress on Image and Signal Processing (CISP), pp. 1–5 (2009)
6. Huang, Y., Benesty, J., Chen, J.: Using the Pearson correlation coefficient to develop an optimally weighted cross relation based blind SIMO identification algorithm. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3153–3156 (2009)
7. Jain, A.K., Ross, A., Prabhakar, S.: An introduction to biometric recognition. IEEE Transaction on Circuit and System for Video Technology 14, 4–20 (2004)
8. Fan, L., Duan, H., Long, F.: Face recognition by subspace analysis of 2D Log-Gabor wavelets features. In: Third International Conference on Intelligent System and Knowledge Engineering(ISKE), pp. 1167–1172 (2008)
9. Zheng, P., Sang, N.: Using Phase and Directional Line Features for Efficient Palmprint Authentication. In: 2nd International Congress on Image and Signal Processing (CISP), pp. 1–5 (2009)
10. Wu, W.J., Xu, Y.: Correlation analysis of visual verb's sub categorization based on Pearson's correlation coefficient. In: International Conference on Machine Learning and Cybernetics (ICMLC), pp. 2042–2046 (2010)
11. PolyU Palmprint database, `http://www4.comp.polyu.edu.hk/~biometrics/`
12. Han, C., Cheng, H., Lin, C., Fan, K.: Personal authentication using palm-print features. Pattern Recognition 36, 371–381 (2003)
13. Funada, J., Ohta, N., Mizoguchi, M., Temma, T., Nakanishi, K., Murai, A., Sugiuchi, T., Wakabayashi, T., Yamada, Y.: Feature extraction method for palmprint considering elimination of creases. In: International Conference on Pattern Recognition, pp. 1849–1854 (1998)

14. Lu, G., Zhang, D., Wang, K.: Palmprint recognition using eigenpalms features. Pattern Recognition Letters 24(9-10), 1463–1467 (2003)
15. Wu, X., Zhang, D., Wang, K.: Fisherpalms based palmprint recognition. Pattern Recognition Letters 24, 2829–2938 (2003)
16. Shang, L., Huang, D.-S., Du, J.-X., Zheng, C.-H.: Palmprint recognition using fast ICA algorithm and radial basis probabilistic neural network. Neurocomputing 69, 1782–1786 (2006)
17. Lu, G., Wang, K., Zhang, D.: Wavelet based independent component analysis for palmprint identification. In: International Conference on Machine Learning and Cybernetics, pp. 3547–3550 (2004)
18. Jing, X., Zhang, D.: A face and palmprint recognition approach based on discriminant DCT feature extraction. IEEE Transactions on Systems, Man, and Cybernetics-B 34, 2405–2415 (2004)
19. Wenxin, L., Zhang, D., Zhuoqun, X.: Palmprint identification by Fourier transform. International Journal of Pattern Recognition and Artificial Intelligence 16, 417–432 (2002)
20. Chen, J., Moon, Y.: Using sift features in palmprint authentication. In: International Conference on Pattern Recognition, pp. 1–4 (2008)
21. Badrinath, G.S., Gupta, P.: Robust biometric system using palmprint for personal verification. In: International Conference on Biometrics, pp. 554–565 (2009)
22. Zhang, D., Kong, A.W., You, J., Wong, M.: Online palmprint identification. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 1041–1050 (2003)
23. Kong, A., Zhang, D.: Competitive coding scheme for palmprint verification. In: International Conference on Pattern Recognition, pp. 520–523 (2004)
24. Daugman, J.: Uncertainty relation for resolution in space, spatial frequency, and oriention optimized by 2D visual cortical filters. Journal of the Optical society of America 2, 1160–1169 (1985)
25. Ross, A., Nandakumar, K., Jain, A.K.: Handbook of Multibiometrics. Springer, New York (2006)

A New Variant of Algebraic Attack

Dibyendu Roy, Pratish Datta, and Sourav Mukhopadhyay

Department of Mathematics,
Indian Institute of Technology Kharagpur,
Kharagpur-721302, India
{dibyendu.roy,pratishdatta,sourav}@maths.iitkgp.ernet.in

Abstract. Algebraic attack is an important attack strategy against symmetric ciphers, particularly stream ciphers. The most vital issue in this attack is to reduce the degree of the algebraic equations as much as possible in order to obtain a lower time complexity. This paper presents one such means of obtaining low degree equations using the decomposition of Boolean functions. This method overcomes the two major drawbacks of fast algebraic attack. We have discussed the general attack strategy using decomposable function. We also demonstrate the decomposition of some Boolean function used in practical stream ciphers. Finally we have given a bound on the degree of a function to be multiplied with a given function so that the product has low degree decomposition.

Keywords: Boolean function, Algebraic Attack, Fast Algebraic Attack, Decomposition of Boolean function.

1 Introduction

In this paper we have considered a LFSR based stream cipher with a non-linear Boolean function. In each clocking the Boolean function takes the state of the LFSR as an input, and it gives the key-stream bits. The state of the LFSR updates by a state update function. Each key-stream bits depends on the the initial state of the LFSR, which is the secret key of the cipher.

By algebraic attack [5] on stream cipher we usually try to find the initial full state of the cipher or some bits of the initial state. In algebraic attack we try to find an algebraic equations between the initial state of the cipher and the key-stream bits. After finding this type of algebraic relation we try to solve this system to get the initial state or some bits of the initial state. There are some methods to solve this system e.g. XL algorithm [4] , Gröbner bases [10], [11], [12] technique. If the algebraic degree of the Boolean function is high then we will get high degree multivariate equations, which is not so easy to solve. Now if we can find low degree equations from high degree equations then solving complexity will be less. In [5] Nicolas Courtois and Willi Meier proved that for any given Boolean function f of n variables we can find a Boolean function g of low degree such that degree of gf becomes low.

There is one variant of algebraic attack, known as fast algebraic attack [3], [1]. In this attack the degree of the equation is reduced in a pre-computation

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 211–222, 2014.

step. Before solving the equations of higher degree, monomials independent of the key-stream bits are eliminated. Here equations of the following form are used $R(L^t \cdot K^0, z_t, ..., z_{t+\delta}) = F(L^t \cdot K) \oplus G(L^t \cdot K, z_t, ..., z_{t+\delta}) = 0$ where F is of degree d of the variable $L^t \cdot K$, G is of degree $e < d$ of the variable $L^t \cdot K$ and key-stream bits. Now the monomials present in F are eliminated by finding a linear combination among the equations such that all the terms in F will be cancelled out. Once the monomials in F are eliminated in the solving step the complexity becomes $o(e^w) \approx o(n^e)$ which is smaller than $o(d^w) \approx o(n^d)$. This method is discussed in detail in section4 . However this method has three important drawbacks e.g., the existence of such a relation R among the state and consecutive key-stream bits for all clocking and the separability of R into high and low degree parts as mentioned above also we need consecutive key-stream bits to construct low degree equations, which is not possible to get for many stream ciphers.

In this paper we have developed a new variant of algebraic attack by using decomposition of Boolean function. If the Boolean function used in the LFSR based stream cipher is decomposable, then we can apply our technique to find low degree multivariate supporting equations from a high degree multivariate equation. How many low degree supporting equations we can find for a high degree equation depends on the decomposed form of the function. Details of this variant is described in section5. After that we have given some examples of stream ciphers on which we can use our technique. After getting these low degree equations we can apply some existing algorithms [10], [11], [4] to solve the system.

2 Basics Study on Boolean Function

Definition 1. ***Boolean function***
A Boolean function f on n variables is a mapping from $\{0,1\}^n$ to $\{0,1\}$.

Definition 2. ***Algebraic Normal Form of Boolean function***
Every Boolean function f can be expressed as a multivariate polynomial over $\mathbb{F}_2$. This polynomial is known as algebraic normal form of the Boolean function f. The general form of algebraic normal form of f is given by,

$$f(x_1, ..., x_n) = a_0 \oplus \bigoplus_{1 \le i \le n} a_i x_i \oplus \bigoplus_{1 \le i < j \le n} a_{ij} x_i x_j \oplus ..... \oplus a_{12...n} x_1 x_2 .... x_n.$$

Definition 3. ***Degree of Boolean function***
Degree of a Boolean function f is defined as $deg(f)$ =Number of variables in the highest order product term in the algebraic normal form of f. Functions of degree at most one are called affine function. An affine function with constant term equal to zero is called linear function.

Definition 4. ***Annihilator of a Boolean function***
A non-zero Boolean function g of n variables is said to be a annihilator of a Boolean function f iff $g(X) \cdot f(X) = 0$, $\forall X \in \{0,1\}^n$.

Following ideas can be found in details in [7].

Definition 5. ***Partition of a Set***
Let us consider two sets $S = \{0,1\}^n$, $V = \{1,2,...,n\}$. Now if for some A_i's $i = 1,2,...,m$, $V = \bigcup_{i=1}^{m} A_i$ and $A_i \cap A_j = \phi$, for $i \neq j$. Then for an element $X \in S$, $X = (x_1, x_2, ..., x_n)$, where for any x_i, there exists one unique A_k, such that $i \in A_k$ where $i = 1,2,....,n$ and $k = 1,2,....,m$.

Definition 6. ***Composition of Boolean functions***
Let us consider few sets, $V = \{1,2,...,n\}$, $I = \{1,2,...,m\}$ and A_i, $i \in I$, and let $V = \bigcup_{i=1}^{m} A_i$ and $A_i \cap A_j = \phi$, for $i \neq j$. $g_i : \{0,1\}^{|A_i|} \to \{0,1\}$ and $F : \{0,1\}^m \to \{0,1\}$ are some defined Boolean functions. Then the Boolean function $f : \{0,1\}^n \to \{0,1\}$ defined by

$$f(X) = F\big(g_1(X_{A_1}), g_2(X_{A_2}), ..., g_m(X_{A_m})\big),$$

is called the composition of the functions F and g_i, $i \in I$.

Definition 7. ***Decomposition of a Boolean function***
For a given Boolean function f if we can find a partition and some Boolean functions F, g_i $i \in I$ such that $f(X) = F\big[g_i, i \in I\big]$ with $|I| > 1$ and $|A_i| \geq 1$ then we say that f is decomposable. And this representation is known as decomposition of f. Otherwise f is indecomposable or prime function.

Definition 8. ***Bi-decomposition***
If the form of the decomposition is $f(X) = F(g(X_A), X_B)$, where the partition is $\{A, B\}$, then we say that f is Bi-decomposable function. And the decomposition is known as Bi-decomposition.

Example 1. Let $f : \{0,1\}^3 \to \{0,1\}$ be Boolean function such that $f(x_1, x_2, x_3) = x_1x_3 \oplus x_2x_3$. Now $f(x_1, x_2, x_3) = x_1x_3 \oplus x_2x_3 = (x_1 \oplus x_2)x_3$. Let $g(x_1, x_2) = x_1 \oplus x_2$ and $F(u, x_3) = ux_3$. Then $f(x_1, x_2, x_3) = F(g(x_1, x_2), x_3)$.

3 Algebraic Attack on LFSR Based Stream Cipher

In this section we will discuss about algebraic attack on LFSRs based stream ciphers. Algebraic attack was introduced by N.T. Courtois[5]. Algebraic attack has two basic strategies-

1. First find a system of algebraic equations involving the secret key using some known key-stream bits.
2. Solve this system to find the secret key.

Now consider a LFSRs based stream cipher with a non-linear function f of n variables. The state update function of the LFSR is L. In each clocking the state of the LFSR is updated by L. Let the initial state i.e. the secret key is $K^0 = (k_0, k_1, ..., k_{n-1})$ of the LFSR . After t-th clocking the state of the LFSR will be $K^t = L^t(K^0)$, where $t \geq 0$. And at t-th clocking the key-stream will be $z_t = f(K^t)$. These bits will be xor-ed with the plain text bits and generate the cipher text bits. Suppose an attacker knows n plain text bits and corresponding n cipher text bits. The attacker can find the corresponding key-stream bits by XORing these known n plain text bits and the corresponding known cipher text bits. Suppose these key-stream bits are $z_{k_1}, z_{k_2}, ..., z_{k_n}$. By using these key-stream bits attacker can construct n algebraic equations over the n unknowns (initial key bits $k_0, k_1, ..., k_{n-1}$). The form of these algebraic system will be as follows,

$$\begin{aligned} f(L^{k_1}(K^0)) &= z_{k_1} \\ f(L^{k_2}(K^0)) &= z_{k_2} \\ f(L^{k_3}(K^0)) &= z_{k_3} \\ &\vdots \\ f(L^{k_n}(K^0)) &= z_{k_n} \end{aligned} \tag{1}$$

Now attacker needs to solve this system to find the initial secret key. Now if the degree of the non-linear function is high then the attacker can multiply f by a low degree annihilator to make the resulting function a low degree one. Then the solution complexity will decrease.

In general the above technique can be alternatively describe as: first find a Boolean function $R \neq 0$ such that $R_t := R(L^t \cdot K^0, z_t, ..., z_{t+\delta}) = 0$ for all clocks t. We will say this equation is a valid equation if it is true for all K^0, t and the corresponding key-stream bits $z_t, ..., z_{t+\delta}$. Using this equation attacker can construct a system of equations over K^0 using known key-stream bits z_t. After that the attacker will solve the resulting system of equations to find secret key K^0. (If we write the equations in terms of the companion matrix then we will get $L^t \cdot K^0$ in place of $L^t(K^0)$.)

4 Fast Algebraic Attack

In this section we will discuss about fast algebraic attack on LFSR based stream ciphers. Fast algebraic attack was first introduced by N.T. Courtois[3] and subsequently modified by Armknecht[1]. Let us consider a LFSRs based stream cipher with a non-linear function f of n variables. It follows the previous technique for generating the key-stream bits. So the key-stream bit after t-th clocking will be $z_t = f(K^t)$. So by the algebraic attack techniques the attacker can find a Boolean function $R \neq 0$ such that

$$R_t := R(L^t \cdot K^0, z_t, ..., z_{t+\delta}) = 0 \tag{2}$$

which is true for all K^0, t and the corresponding key-stream bits. Fast algebraic attack works *iff* the equation (2) can be rewritten as

$$R(L^t \cdot K^0, z_t, ..., z_{t+\delta}) = F(L^t \cdot K^0) \oplus G(L^t \cdot K^0, z_t, ..., z_{t+\delta}) = 0 \qquad (3)$$

where $deg_{K^0}(G) < deg_{K^0}(R)$. After that attacker needs to find coefficients $\lambda_0, ..., \lambda_{T-1} \in \{0, 1\}$ such that

$$\bigoplus_{i=0}^{T} \lambda_i \cdot F(L^{t+i} \cdot K^0) = 0 \quad \forall t, K^0 \qquad (4)$$

The coefficients λ_i's are independent of K^0. Now from (3) and (4) we have

$$\bigoplus_{i=0}^{T} \lambda_i R_{t+i} = \bigoplus_{i=0}^{T} \lambda_i \cdot G(L^{t+i} \cdot K^0, z_{t+i}, ..., z_{t+i+\delta}) = 0 \qquad (5)$$

This is an valid low degree equation. By repeating above procedure for several clocks t, the attacker can construct system of low degree equations from a system of high degree equations. The system will be

$$\left.\begin{array}{l} 0 = R(K^0, z_0, ..., z_\delta) \\ 0 = R(L \cdot K^0, z_1, ..., z_{\delta+1}) \\ .... \\ \textit{high degree equations} \end{array}\right\} \Rightarrow \left\{\begin{array}{l} 0 = \bigoplus_{i=0}^{T} \lambda_i \cdot G(L^i \cdot K^0, z_i, ..., z_{i+\delta}) \\ 0 = \bigoplus_{i=0}^{T} \lambda_i \cdot G(L^{1+i} \cdot K^0, z_{1+i}, ..., z_{1+i+\delta}) \\ .... \\ \textit{low degree equations} \end{array}\right.$$

Hence the attacker is getting low degree equations, which is less complex than the previous one. This method works quite well for the three ciphers E_0, Toyocrypt, and LILI-128.

However, fast algebraic attack, in general, is based on three strong assumptions as follows-

1. The relation R remains same for all clocking.
2. R should be separable into high and low degree parts where the high degree part has to be independent of the key-stream bits.
3. To construct the low degree equations in fast algebraic attack we need to take consecutive key-stream bits.

These three requirements make this attack applicable to only a restricted class of stream ciphers. Our method, as discussed in the next section does not require any such universal relation. Also we have considered the decomposition of Boolean functions in more general way. These two modifications are certain to make our method applicable to a larger class of stream ciphers.

5 New Variant of Algebraic Attack

In this section we will discuss our new attack strategy. Consider a LFSR based stream cipher with a non-linear function f of n variables. The state of the LFSR is updated by the state update function L. The non-linear function takes the state of LFSR as input and produces the key-stream bits. By using the same technique used in algebraic attack we can find the same algebraic system of equation given in (1). We need to solve the algebraic system (1) to get the initial key K^0.

Suppose the function f is decomposable i.e. there exists a partition and some Boolean functions $F, g_i \; i \in I$ such that $f(X) = F\big[g_i, i \in I]$ with $|I| > 1$ and $|A_i| \geq 1$. For simplicity of discussion we are assuming that the function f is bi-decomposable i.e. there exists a partition $\{A, B\}$ and two functions F and g such that $f(X) = F(g(X_A), X_B)$. Where $g : \{0,1\}^k \to \{0,1\}$ and $F : \{0,1\}^{m+1} \to \{0,1\}$ where $k + m = n$.

As g and F acts on less numbers of variables, so the $deg(g), deg(F) < deg(f)$.

Now, we will describe what will be the attack scenario:

Consider the first equation $f(L^{k_1}(K^0)) = z_{k_1}$. Let $L^{k_1}(K^0) = X^{k_1}$. As f is bi-decomposable then this equation can be rewritten as $F(g(X_A^{k_1}), X_B^{k_1}) = z_{k_1}$.

$$\begin{aligned} & f(L^{k_1}(K^0)) = z_{k_1} \\ \Rightarrow\ & F(g(X_A^{k_1}), X_B^{k_1}) = z_{k_1} \\ \Rightarrow\ & \left\{ \begin{array}{l} F(u^{k_1}, X_B^{k_1}) = z_{k_1} \\ u^{k_1} = g(X_A^{k_1}) \end{array} \right. \end{aligned}$$

So, from one high degree equation we are getting two low degree equations on less number of variables. Similarly for other equations we can apply same procedure to get two low degree simultaneous equations. So, the previous system of equations of high degree reduces to low degree simultaneous equations.

$$\left. \begin{array}{l} f(L^{k_1}(K^0)) = z_{k_1} \\ f(L^{k_2}(K^0)) = z_{k_2} \\ \dots\dots\dots\dots\dots\dots \\ \dots\dots\dots\dots\dots\dots \\ \dots\dots\dots\dots\dots\dots \\ f(L^{k_n}(K^0)) = z_{k_n} \\ \\ \textit{high degree equations} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \left\{ \begin{array}{l} F(u^{k_1}, X_B^{k_1}) = z_{k_1} \\ u^{k_1} = g(X_A^{k_1}) \end{array} \right. \\ \left\{ \begin{array}{l} F(u^{k_2}, X_B^{k_2}) = z_{k_2} \\ u^{k_2} = g(X_A^{k_2}) \end{array} \right. \\ \quad \dots\dots\dots\dots\dots\dots \\ \quad \dots\dots\dots\dots\dots\dots \\ \quad \dots\dots\dots\dots\dots\dots \\ \left\{ \begin{array}{l} F(u^{k_n}, X_B^{k_n}) = z_{k_n} \\ u^{k_n} = g(X_A^{k_n}) \end{array} \right. \\ \\ \textit{low degree simultaneous equations} \end{array} \right.$$

As in each clocking the algebraic normal form of non-linear function f remains same, only the variables are changing. So, the structure of the decomposition of the function f remains same for all clocking.

Consider a small example where we will get low degree equations form one high degree equation by using our technique.

Example 2. Consider a four bit LFSR with the non-linear Boolean function $f(x_1, x_2, x_3, x_4) = x_1x_3 \oplus x_1x_4 \oplus x_1x_2x_3 \oplus x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_1x_2x_3x_4$. The state update function $L(x_1, x_2, x_3, x_4) = x_2 + x_3$. Clearly $f(x_1, x_2, x_3, x_4) = F(g_1(x_1, x_2), g_2(x_3, x_4))$ where $g_1(x_1, x_2) = x_1x_2 \oplus x_1$, $g_2(x_3, x_4) = x_3 + x_4 + x_3x_4$ and $F(u, v) = uv$. For the first key-stream bit

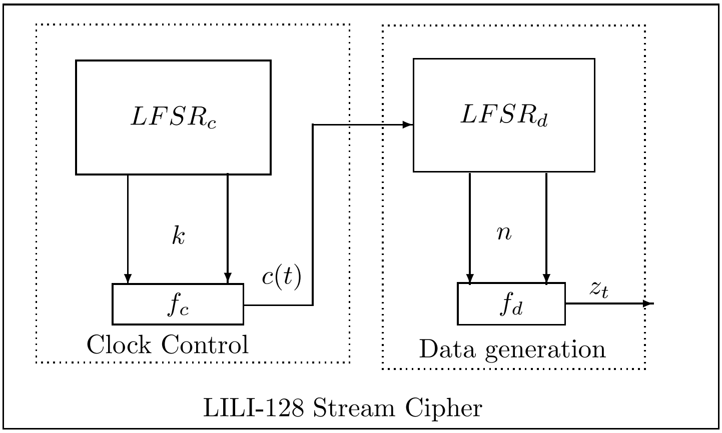
$$x_1x_3 \oplus x_1x_4 \oplus x_1x_2x_3 \oplus x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_1x_2x_3x_4 = z_0$$
$$\Rightarrow F(g_1(x_1, x_2), g_2(x_3, x_4)) = z_0$$
$$\Rightarrow \begin{cases} uv = z_0 \\ u = x_1x_2 \oplus x_1 \\ v = x_3 + x_4 + x_3x_4 \end{cases}$$

So, we are getting two second degree simultaneous equations instead of one third degree equation. Now if $z_0 = 1$ then $u = 1$ and $v = 1$. So by using our technique instead of $x_1x_3 \oplus x_1x_4 \oplus x_1x_2x_3 \oplus x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_1x_2x_3x_4 = 1$ this equation we are getting two equations of the form $x_1x_2 \oplus x_1 = 1$ and $x_3 + x_4 + x_3x_4 = 1$ which are of degree 2 and lesser than the degree of the original equation. Now from $x_1x_2 \oplus x_1 = 1$ we are getting $x_1(1 \oplus x_2) = 1$. From this equation we are getting $x_1 = 1, x_2 = 0$. Similarly for other key-stream bits we can rewrite the equations into second degree equations and can substitute the values of x_1, x_2 . So, by using our technique we are able to construct some low degree equations which are easier to solve than the exact equations.

Now, we will discuss the decomposed form of the Boolean functions used in some practical stream ciphers.

5.1 Decomposed Form of the Non-linear Boolean Function in LILI-128 Stream Cipher

The description of LILI-128 stream cipher[9] is given in the following figure



The primitive polynomial for LFSR$_c$ is

$$g_c(x) = x^{39} + x^{35} + x^{33} + x^{31} + x^{17} + x^{15} + x^{14} + x^2 + 1$$

The clock control function

$$c_k = f_c(y_1, y_2) = 2y_1 + y_2 + 1,\ k \geq 1$$

The primitive polynomial of LFSR$_d$ is

$$g_d(x) = x^{89} + x^{83} + x^{80} + x^{55} + x^{53} + x^{42} + x^{39} + x + 1$$

The nonlinear function f_d is of 10 variables 6^{th} degree.

$$\begin{aligned}
&f_d(x_1,..,x_{10}) = x_2 + x_3 + x_4 + x_5 + x_6x_7 + x_1x_8 + x_2x_8 + x_1x_9 + x_3x_9 + x_4x_{10} \\
&+ x_6x_{10} + x_3x_7x_9 + x_4x_7x_9 + x_6x_7x_9 + x_3x_8x_9 + x_6x_8x_9 + x_4x_7x_{10} + x_5x_7x_{10} \\
&+ x_6x_7x_{10} + x_3x_8x_{10} + x_4x_8x_{10} + x_2x_9x_{10} + x_3x_9x_{10} + x_4x_9x_{10} + x_5x_9x_{10}+ \\
&x_3x_7x_8x_{10} + x_5x_7x_8x_{10} + x_2x_7x_9x_{10} + x_4x_7x_9x_{10} + x_6x_7x_9x_{10} + x_1x_8x_9x_{10}+ \\
&x_3x_8x_9x_{10} + x_4x_8x_9x_{10} + x_6x_8x_9x_{10} + x_4x_6x_7x_9 + x_5x_6x_7x_9 + x_2x_7x_8x_9+ \\
&x_4x_7x_8x_9 + x_4x_6x_7x_9x_{10} + x_5x_6x_7x_9x_{10} + x_3x_7x_8x_9x_{10} + x_4x_7x_8x_9x_{10}+ \\
&x_4x_6x_7x_8x_9 + x_5x_6x_7x_8x_9 + x_4x_6x_7x_8x_9x_{10} + x_5x_6x_7x_8x_9x_{10}
\end{aligned}$$

Now if we multiply $f_d(x_1, x_2, ..., x_{10})$ by x_8x_{10}, we will get

$$f_d(x) \cdot x_8x_{10} = x_8x_{10}[x_1 + x_4 + x_5 + x_6 + x_2x_9 + x_4x_7 + x_3x_7 + x_5x_9]$$

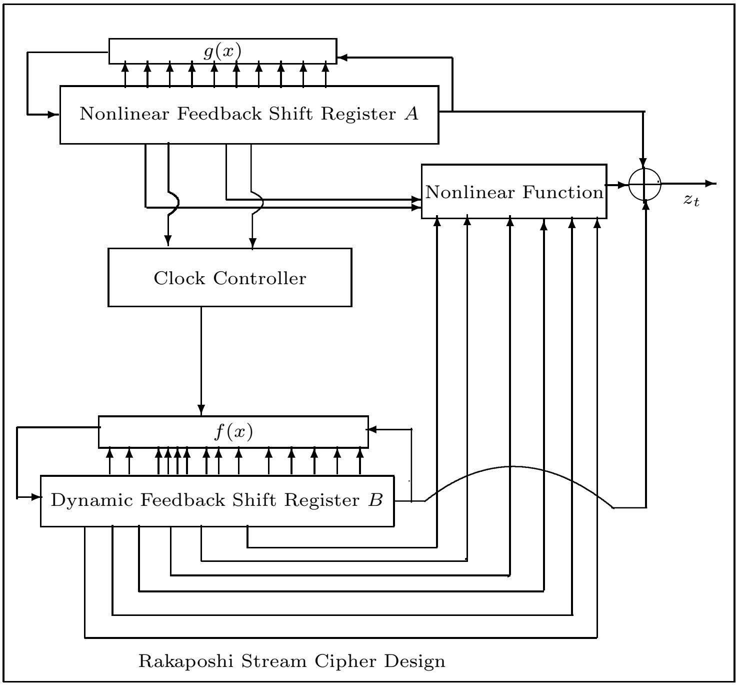
Now

$$\begin{aligned}
f_d(x) \cdot x_8x_{10} &= x_8x_{10}[x_1 + x_4 + x_5 + x_6 + x_2x_9 + x_4x_7 + x_3x_7 + x_5x_9] \\
&= x_8x_{10}[(x_1 + x_6) + (x_4 + x_5 + x_2x_9 + x_4x_7 + x_3x_7 + x_5x_9)] \\
&= \begin{cases} F(u_1, x_1, x_6, u_2) = u_1[x_1 + x_6 + u_2] \\ u_1 = g_1(x_8, x_{10}) = x_8x_{10} \\ u_2 = g_2(\cdot) = x_4 + x_5 + x_2x_9 + x_4x_7 + x_3x_7 + x_5x_9 \end{cases}
\end{aligned}$$

From a sixth degree Boolean function we are getting second degree simultaneous Boolean functions. Also the Boolean functions are on less number of variables. So, the form of the degree of the algebraic equations of key-streams for this stream cipher will be lesser than the previous one.

5.2 Decomposed Form of the Non-linear Boolean Function in Rakaposhi Stream Cipher

The description of Rakaposhi stream cipher[2] is given in the following figure


$g(x)$
Nonlinear Feedback Shift Register A
Nonlinear Function
z_t
Clock Controller
$f(x)$
Dynamic Feedback Shift Register B
Rakaposhi Stream Cipher Design

Non-linear Feedback Shift Register A: The NLFSR is of 128 bits. The feedback function is

$$\begin{aligned} g(x_0, x_1, ..., x_9) =& x_1x_3x_9 + x_1x_7x_9 + x_5x_8 + x_2x_5 + \\ & x_3x_8 + x_2x_7 + x_9 + x_8 + x_7 + x_6 + \\ & x_5 + x_4 + x_3 + x_2 + x_1 + x_0 + 1 \end{aligned}$$

Linear Feedback Shift Register B: The Dynamic LFSR is of 192 bits. The feedback function is

$$\begin{aligned} f(x) =& x^{192} + x^{176} + c_0x^{158} + (1 + c_0)x^{155} + c_0c_1x^{136} + \\ & c_0(1 + c_1)x^{134} + c_1(1 + c_0)x^{120} + (1 + c_0)(1 + c_1)x^{107} + \\ & x^{93} + x^{51} + x^{49} + x^{41} + x^{37} + x^{14} + 1. \end{aligned}$$

Where c_0 and c_1 42^{nd} and 90^{th} bits of the NLFSR A at t-th clocking.

The non-linear function $v(\cdot)$ is of 8 variables 7^{th} degree. The description of the non-linear function is given in [2]. Now, if we multiply $v(\cdot)$ by $x'_0x'_1x'_2x'_3$, we will get

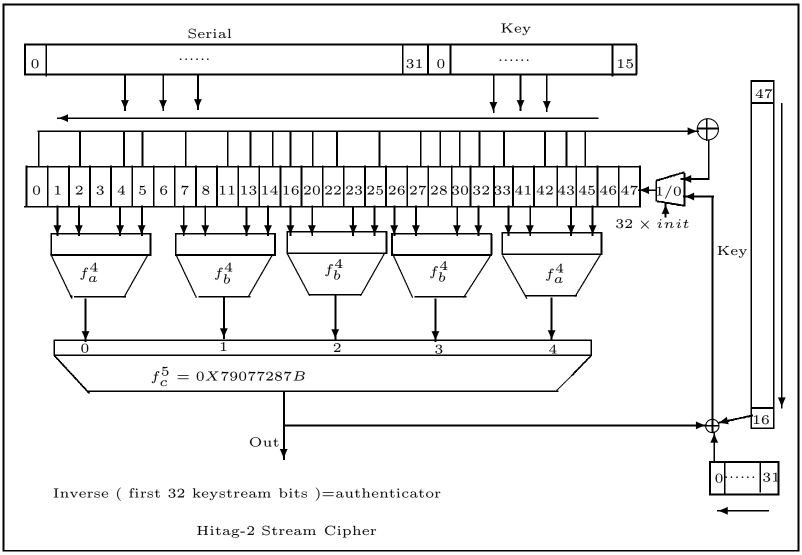
$$\begin{aligned} v(\cdot)x'_0x'_1x'_2x'_3 &= x'_0x'_1x'_2x'_3[x_4x_5x_6 + x_4x_5 + x_5x_6x_7 + x_5x_6 + x_5 + x_6 + x_7] \\ &= F(g(x_0, x_1, x_2, x_3), x_4, x_5, x_6, x_7) \end{aligned}$$

Where $g(x_0, x_1, x_2, x_3) = (1+x_0)(1+x_1)(1+x_2)(1+x_3)$ and $F(u, x_4, x_5, x_6, x_7) = u[x_4x_5x_6 + x_4x_5 + x_5x_6x_7 + x_5x_6 + x_5 + x_6 + x_7]$. So, from a 7^{th} degree Boolean function we are getting two simultaneous Boolean functions of 4^{th} degree.

Remark. Initially the non-linear function was of 7^{th} degree and the degree of the NLFSR feedback function was 3. For this reason the degree of the algebraic expressions of the consecutive key-stream output bits were increasing very rapidly. But after decomposing the non-linear function we are basically getting two 4^{th} degree simultaneous Boolean functions. That is why the degree of the algebraic expression of the successive key-stream output bits will now increase at a lesser rate. This definitely improve the complexity of the algebraic attack.

5.3 Decomposed Form of the Non-linear Boolean Functions in Hitag-2 Stream Cipher

The description of Hitag-2 stream cipher[6] is given in the following figure



This stream cipher uses a LFSR is of 48 bits and non-linear function that takes 20 inputs and gives single bits output. The non-linear functions used in the ciphers are

$$f_a^4 = 0X2C79 = abc + ac + ad + bc + a + b + d + 1$$
$$f_b^4 = 0X6671 = abd + acd + bcd + ab + ac + bc + a + b + d + 1$$

If we multiply f_a^4 by ad then we will get $f_a^4 \cdot ad = ad(c + b)$. So we have two simultaneous Boolean functions of degree 2.

$$\begin{cases} f_a^4 \cdot ad = u_1(c + b) \\ u = ad \end{cases}$$

Similarly if we multiply f_b^4 by bc then we will get $f_b^4 \cdot bc = bc(1 + a)$. Then we have two simultaneous Boolean functions of degree 2.

$$\begin{cases} f_b^4 \cdot bc = u_2(1 + a) \\ u_2 = bc \end{cases}$$

Algebraic attacks on this cipher has been done by Nicolas T. Courtois et al[6]. By using this decomposition of the Boolean functions we will get less degree and less complex algebraic expression for the key-stream bits. So, it will decrease the complexity of the algebraic attack and also increase the speed of the attack.

So by previous examples we can see that, given a Boolean function f the function is either in decomposed form or we can find some Boolean function g such that $g \cdot f$ is in decomposed form.

Theorem 1. *For any Boolean function $f : \{0,1\}^n \to \{0,1\}$, there exists a function $h : \{0,1\}^{\lceil \frac{n}{2} \rceil} \to \{0,1\}$ of degree at most $\lceil \frac{n}{2} \rceil$ such that $h \cdot f$ can be decomposed using functions of degree at most $\lceil \frac{n}{2} \rceil$.*

Proof. So $f : \{0,1\}^n \to \{0,1\}$ is a Boolean function of n variables. Consider $h(x_1, x_2, ..., x_{\lceil \frac{n}{2} \rceil}) = x'_1 x'_2 .... x'_{\lceil \frac{n}{2} \rceil}$. Then if we multiply f by this h, all the terms in f involving at least one of the variables $x'_1, x'_2, ..., x'_{\lceil \frac{n}{2} \rceil}$ will vanish. After this multiplication we will get, the terms in f involving only the remaining variables multiplied by h. So the general form of $h.f$ will be $h.g(x_{\lceil \frac{n}{2} \rceil+1}, x_{\lceil \frac{n}{2} \rceil+2}, ..., x_n)$. Now we can define a Boolean function $t : \{0,1\}^2 \to \{0,1\}$ such that $t(y_1, y_2) = y_1 y_2$. Now taking $y_1 = h$ and $y_2 = g$ we will get $t(y_1, y_2) = h.f$. Clearly $h.f$ is now in decomposed form. As h is function of the variables $x_1, x_2, ..., x_{\lceil \frac{n}{2} \rceil}$, so degree of h can be at most $\lceil \frac{n}{2} \rceil$. And also note that g is a function of $n - \lceil \frac{n}{2} \rceil (\le \lceil \frac{n}{2} \rceil)$ variables. So the degree of g will be at most $\lceil \frac{n}{2} \rceil$. This proves the existence of h.

From the above theorem the next corollary follows directly.

Corollary 1. *If the degree of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is strictly greater than $\lceil \frac{n}{2} \rceil$ then there exists a function h of degree less than or equal to $\lceil \frac{n}{2} \rceil$ such that hf has low degree decomposition.*

6 Conclusion

In this paper we have introduced the idea of obtaining simultaneous low degree multivariate equations from one high degree equation. Note that if the non-linear function f can be decomposed as $f = F(g_i),\ i = 1, 2, ..., m$ then we get $m+1$ low degree equations from one high degree equation. Thus the size of resulting system has increased but it is known that solving a larger system of lower degree can be more efficient than that of a relatively smaller system of higher degree. However for many functions we can have more than one low degree decomposition and depending on different decomposition we can obtain corresponding equivalent system of various sizes. The trade off between these two should be appropriately taken care of to improve the efficiency of the attack further. Also it remains open question that, if the function f has degree $< \lceil \frac{n}{2} \rceil$ whether we can obtain a low degree decomposition or not. It is certain that our method can be applied for stream cipher on which fast algebraic attack may not be applicable. But if we apply this new method to ciphers on which fast algebraic attack is applicable, whether it gives a better time complexity or not also needs a thorough analysis.

References

1. Armknecht, F.: Improving fast algebraic attacks. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 65–82. Springer, Heidelberg (2004)
2. Cid, C., Kiyomoto, S., Kurihara, J.: The rakaposhi stream cipher. Information and Communications Security, 32–46 (2009)
3. Courtois, N.T.: Fast algebraic attacks on stream ciphers with linear feedback. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 176–194. Springer, Heidelberg (2003)
4. Courtois, N.T., Klimov, A.B., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer, Heidelberg (2000)
5. Courtois, N., Meier, W.: Algebraic attacks on stream ciphers with linear feedback. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 345–359. Springer, Heidelberg (2003)
6. Courtois, N., O'Neil, S., Quisquater, J.J.: Practical algebraic attacks on the hitag2 stream cipher. In: Information Security, pp. 167–176 (2009)
7. Crama, Y., Hammer, P.L.: Boolean models and methods in mathematics. In: Computer Science and Engineering (2010)
8. Cusick, T.W., Stănică, P.: Cryptographic Boolean functions and applications. Academic Press (2009)
9. Dawson, E., Clark, A., Golic, J., Millan, W., Penna, L., Simpson, L.: The lili-128 keystream generator. In: Proceedings of first NESSIE Workshop (2000)
10. Faugre, J.C.: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra 139(1-3), 61–88 (1999), `http://www-salsa.lip6.fr/~jcf/Papers/F99a.pdf`
11. Faugre, J.C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In: International Symposium on Symbolic and Algebraic Computation, pp. 75–83 (2002), `http://www-salsa.lip6.fr/~jcf/Papers/F02a.pdf`
12. Segers, A.: Algebraic attacks from a gröbner basis perspective. Master's Thesis (2004)

A Novel Design of Chaos Based S-Box Using Difference Distribution Table (CD S-Box)

Muhammad Asif Khan and Varun Jeoti

Department of Electrical and Electronic Engineering,
Universiti Teknologi PETRONAS, Darul Ridzuan, 31750 Tronoh, Perak Malaysia
asif48@gmail.com, Varun_jeoti@petronas.com.my

Abstract. This research work reports the design methodology of a novel chaotic substitution box that is dynamically designed by systematically optimizing using DDT. DDT is a tool that helps in differential cryptanalysis of S-box. The proposed S-box shows very low differential probability as compared to other chaos based deigned S-box recently, while maintaining good cryptographic properties and linear approximation probability. Our proposed CD S-box achieves very low differential approximation probability of 8/256.

Keywords: Substitution box, chaos, differential cryptanalysis.

1 Introduction

In a typical cryptosystem, substitution box (S-box) is the only nonlinear component that helps implement confidentiality and plays a vital role in determining the security of cryptosystems. S-box is deployed in almost all conventional cryptosystems, such as DES, DES like cryptosystems and AES. ‘Confusion’ and ‘diffusion’, considered fundamental for designing modern cryptographic algorithms, were first formulated by Shannon in his great seminal paper in 1949 [1].

In the past decade, much attention has been focused on chaos based design. It attracts attention because chaotic orbits are boundedly aperiodic, unpredictable and sensitive to initial conditions. Researchers [2-5] find remarkable similarities between chaos and cryptography, therefore, chaos is considered an alternative to design secure S-boxes that is deployed in cryptosystems. Jakimoski and Kocarev [4] first presented chaos based S-box design. Afterwards, it is extended from 1D-maps to higher order discretized chaotic maps for improved S-box design [6-12].

Chaos based design methods are undertaken for our study. Chaos based methods can improve cryptographic properties to a certain limit, specially linear and differential probabilities. With these challenges ahead and strong cryptographic attacks (mainly linear and differential attacks) systematic design of S-box optimization would be paramount to achieve the desirable performance. Towards this end, a number of techniques have been proposed to optimize S- box to achieve near optimal properties in terms of higher nonlinearity [13-16]. Recently, optimization methods have been merged with chaos based techniques to optimize S-box, doing so appears to have good cryptographic properties as compared to optimization techniques (namely

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 223–230, 2014.

genetic algorithms, evolutionary methods and search based techniques) or chaos based design [17]. Incorporating chaos based design with optimization methods can improve desired properties set that is chosen to optimize. However, it seems challenging to optimize all cryptographic properties at once. For example, there is not much improvement for linear and differential probabilities using chaos based design of S-box. Moreover, to the best of our knowledge, not a single systematic design method is available in literature to improve the linear and differential probabilities.

In this work, a novel chaotic systematic S-box is presented. The objective is to minimize differential probability. Towards this end, the DDT has been used as a tool to verify the resistance of S-box against differential attack dynamically. The DDT has never been considered for designing S-box. Previously, S-box has been designed and optimized against one or more cryptographic properties while differential probabilities are retained high in random based design. To the best of our knowledge, random based design improvement is accidental and not systematic, because algorithms are not optimized to achieve near optimal cryptographic properties and minimal differential probabilities. Rather these properties are achieved from random source to generate S-box positions. However, in this work, S-box is designed dynamically using DDT to achieve low differential probability and keeping near optimal cryptographic properties.

In the remainder of this paper, following organization of the material is adhered to. In section 2 introducing preliminaries, we briefly introduce differential and linear probabilities, the DDT and Logistics map. In section 3, we present the algorithm itself and follow that up with its evaluation in section 4.

2 Preliminaries

2.1 Differential and Linear Probabilities

The concept of differential cryptanalysis [18] was introduced by Eli Biham and Adi Shamir in 1990. The differential cryptanalysis seeks to find any structural weakness in the given cryptosystem. These weaknesses are highlighted by analyzing the S-box for differential approximation probabilities (DP) that measures the probable differential characteristics. Differential characteristic measures the occurrence of highest number of output difference pairs whose input pairs have a particular difference and is defined as:

$$\mathrm{DP}(\Delta y \to \Delta x) = \left(\frac{\#\{x \in X | S(x) \oplus S(x \oplus \Delta x) = \Delta y\}}{2^N}\right) \tag{1}$$

where X denotes all possible input values and 2^N denotes total number of elements. In general, DP is the probability of having Δy, when the input difference is Δx.

Another important concept of linear cryptanalysis [19] was introduced by Mitsuru Matsui in 1993. It measures the linear approximation by XOR input bits together, XOR output bits together and XOR the input and output bit. Finally, XOR it with the key bits measures the linear approximation probability. It measures the maximum imbalance of an event: the parity of the input bits selected by the mask Γ x is equal to the parity of the output bits selected by the mask Γy. The linear approximation probability is defined as:

$$\mathrm{LP}(\Gamma y \to \Gamma x) = \max_{\Gamma x, \Gamma y \neq 0} \left| \left(\frac{\#\{x | x \bullet \Gamma x = s(x) \bullet \Gamma y\}}{2^N} \right) - \frac{1}{2} \right| \quad (2)$$

where Γx and Γy are the input and output masks respectively, X denotes the set of all possible inputs and 2^N is the number of its elements.

The differential cryptanalysis is theoretically a strong attack in nature and requires large space to search for the pairs. Therefore, it might eventually increase the time complexity to mount an attack [20]. To minimize this, few heuristic techniques are proposed. In [21], author proposed a neural network model to find differential characteristics to represent the differential operation for block encryption.

2.2 Introduction to Difference Distribution Table (DDT)

The difference distribution table is formed to measure the maximum differential probability of S-box. The purpose is to analyze the level of resistance an S-box provides against differential cryptanalysis. The differential cryptanalysis seeks high probability of occurrences of output difference pairs whose corresponding input pairs have particular difference. The input to an S-box is referred as $X = [X_1, X_2, X_3 \dots X_N]$ and output is referred as $Y = [Y_1, Y_2, Y_3 \dots Y_N]$. Each input and output to an S-box is comprises of N bits. As a result, the total number of inputs an S-box can entertain would be 2^N. The input difference is denoted as $\Delta X = X' \oplus X''$ and corresponding output difference is denoted by $\Delta Y = Y' \oplus Y''$. The S-box values (X', X'') and (Y', Y'') are the input and out pairs, where " $\oplus$ " represents the bit-wise exclusive-OR. The input differences are in the range of $[1, n = 2^N]$. For example, as described in figure 1, we have given inputs X' and X'' with a difference of '1' to our S-box, which substitutes them to output Y' and Y'' thus result in output difference of $\Delta Y = Y' \oplus Y'' = n$. The pair $(\Delta X, \Delta Y)$ is called a differential pair. DDT possesses several properties. The rows of DDT represent input differences ΔX, while column represents output differences ΔY. The sum of output differences in a single row or in single column of DDT is 2^N. All output differences ΔY in each DDT column have even values because they occur in pairs. For example, input difference ΔX is $\Delta X = X' \oplus X'' = X'' \oplus X'$. Moreover, input difference of $\Delta X = 0$ leads to output difference of $\Delta Y = 0$ for a bijective S-box. Hence, the first element of first column is 8 and other values in first row and column are 0. For an ideal S-box that gives no information about output differential, all elements of DDT have the value of 1. Therefore, the probability of occurrence of an output difference for given input difference is $\frac{1}{2^n} = \frac{1}{2^3} = \frac{1}{8}$. However, it is not achievable because the differentials always occur in pairs.

2.3 Chaotic Logistic Map

The proposed S-Box starts with an initial S-Box which is systematically optimized for better differential probability. This initial S-Box is derived from using a chaotic logistic map. The reason for using chaos is due to its inherent properties that can generate highly random positions. The proposed methodology of the S-BOX design will be explained in later section.

The Chaotic logistic map is a well known 1-D chaotic map and it is simple to implement.

Chaotic logistic map is defined as (3)

$$x_{n+1} = rx_n(1 - x_n) \tag{3}$$

where $0 \leq x_n \leq 1$ and $2.57 < r \leq 4$

By iterating the chaotic logistic map with a unique initial value $0 < x_0 < 1$ one can generate a unique sequence of random real numbers whose values lie between 0 and 1. Chaotic logistic map is used with r = 4, is the only useful case in equation 3 because the chaotic attractor are distributed uniformly in chaotic domain region which is span over [0, 1].

3 The Algorithm

The proposed methodology to systematically design S-box with low differential probability is two-fold. First, initial pool positions for optimization are generated using chaotic logistic map. Secondly, DDT analysis is used to systematically check and if necessary regenerate the positions in order that these positions prove optimal choice for final S-box.

Our aim is to design bijective S-box, therefore we generate substitution matrix 'S'

$$S = \{S_i, \quad S_{i+1}, \quad \dots \quad S_N\}^T \tag{4}$$

where s_i in substitution matrix is a row vector with $N-1$ zeros and only one 1. Substitution matrix S is of size N×N where N denotes the number of input elements to be substituted. The position of '1' in row vector s_i denotes the position of ith input element after substitution.

$$S_i(P_i) = 1 \tag{5}$$

where P_i denote the position of '1' in row vector S_i.

The initial position vector P is generated by iterating chaotic logistic map. The position vector in turn determines the elements of the initial S-BOX. However, to design the proposed S-box, DDT is used to systematically optimize the positions of this initial S-Box. The DDT helps in achieving low differential probability which is reflected in the observation that, for a given input difference, the output differences in DDT columns of S-box are fairly dispersed and equally distributed.

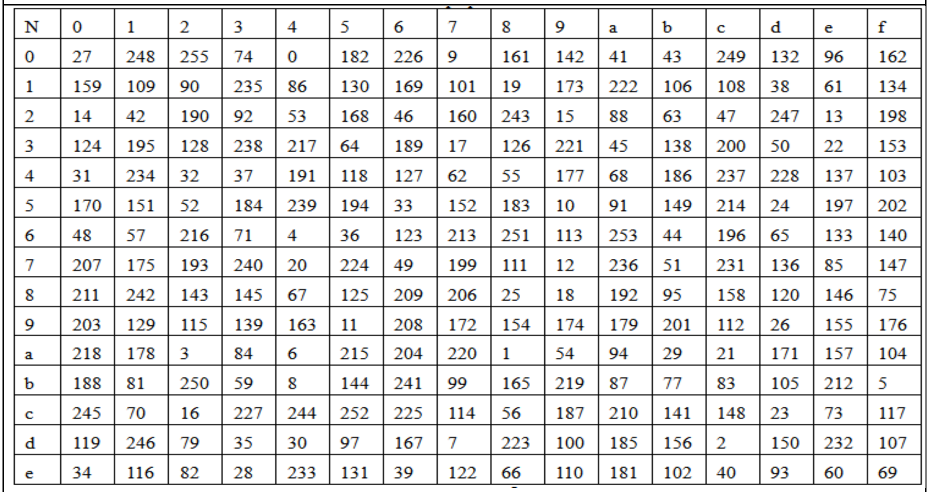
The detail of proposed algorithm is given in pseudo code. Moreover, detail flow chart is also given in Fig. (2). Here by, the summary of proposed scheme is given.

To design our proposed S-box, the domain in the range of [0.1, 0.9] is divided into n equal intervals. These intervals are then labeled sequentially. The logistic map is iterated with a random initial condition. The output of logistic map is checked in the domain to mark where it falls and corresponding subdomain number is stored in row vector which is called as position vector P. During the course of iteration, if the map output falls in a visited subdomain then this subdomain is ignored. As we are designing the S-box dynamically, thus first we need to generate and store first two positions in position vector. The difference between generated positions are calculated and checked in DDT columns, that represents input difference, for difference repetition. If difference is repeated in any column of DDT then this particular position is ignored and is generated again. Otherwise, it retains in position vector and procedure continues to generate and check next position. For each subsequent position, the difference is computed with prior consecutive positions in the position vector. In the situation of deadlock where generated position cannot retain in specific DDT column without increment in difference repetition, then repetition in each column is allowed one time. The given procedure is repeated until all S-box positions are generated.

3.1 The Resulting CD S-Box

Here by, an example of a typical S-Box resulting from using the proposed algorithm is shows in Table 1.

Table 1. Proposed CD S-box



N	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	27	248	255	74	0	182	226	9	161	142	41	43	249	132	96	162
1	159	109	90	235	86	130	169	101	19	173	222	106	108	38	61	134
2	14	42	190	92	53	168	46	160	243	15	88	63	47	247	13	198
3	124	195	128	238	217	64	189	17	126	221	45	138	200	50	22	153
4	31	234	32	37	191	118	127	62	55	177	68	186	237	228	137	103
5	170	151	52	184	239	194	33	152	183	10	91	149	214	24	197	202
6	48	57	216	71	4	36	123	213	251	113	253	44	196	65	133	140
7	207	175	193	240	20	224	49	199	111	12	236	51	231	136	85	147
8	211	242	143	145	67	125	209	206	25	18	192	95	158	120	146	75
9	203	129	115	139	163	11	208	172	154	174	179	201	112	26	155	176
a	218	178	3	84	6	215	204	220	1	54	94	29	21	171	157	104
b	188	81	250	59	8	144	241	99	165	219	87	77	83	105	212	5
c	245	70	16	227	244	252	225	114	56	187	210	141	148	23	73	117
d	119	246	79	35	30	97	167	7	223	100	185	156	2	150	232	107
e	34	116	82	28	233	131	39	122	66	110	181	102	40	93	60	69

4 Performance Analysis of the Proposed S-Box

To evaluate the performance of proposed S-Box design, all performance criterions are analyzed in detail. Here, we list all essential cryptographic properties which are widely accepted in literature and used for performance evaluation [10, 17, 22-24]. It is shown that proposed scheme has very low linear and differential probabilities as compared to randomly and optimized S-box to date.

4.1 Differential Approximation Probability (DP)

The nonlinear mapping or S-box should ideally have differential uniformity. The differential approximation probability for a given S-box, DP in short, is measure for differential uniformity and defined and discussed in equation (1).

The objective of this work is to optimize differential probability, which is never targeted before for optimization using DDT, and make it as low as possible. Fig (2) shows the distribution of differentials generated using our proposed method, where x-axis denote the input difference Δx, whereas y-axis represents the output difference Δy and z-axis shows the number of occurrences of the particular input and output differential $(\Delta x, \Delta y)$. The differential of our proposed S-box occurs with the maximum probability of $\frac{8}{256}$. Moreover, the probability of differentials occurs with $\frac{8}{256}$ is very low. Most of the entries in DDT are 0, 2, 4 and 6. As compared to recently published S-box where all of the entries in the range of 6, 8, 10, and in some cases 12. For comparison, the DPs of proposed S- box and other recently proposed S-boxes are listed in table (2). The other S-boxes have higher maximum DPs of $\frac{10}{256}$ and $\frac{12}{256}$ as compared to our proposed work, and not a single randomly designed or randomly optimized S-box achieve this low probability to date. The histogram analysis is also carried out. The histogram of differentials is shown in figure 1. The entire differentials are uniformly distributed. It is evident from the histogram that very few differentials have the probability of 8. Rest of the differential are occurred with very low probabilities.

4.2 Linear Approximation Probability (LP)

Linear Approximation probability (or probability of bias) of a given S-box, according to Matsui's original definition [19], is defined and discussed in equation (2) . The LP of proposed S-box and recently published S-boxes is compared according to eq. (12) and listed in table (3). The LP of proposed S-box is 0.1094, which is very low and shows better performance as compared to other S-boxes and well satisfies the cryptographic criteria of LP.

Table 2. A Comparison of DP

S-box	Maximum DP
Proposed S-box	8/256
Asim et al.[22]	10/256
Hussain et al. [23]	10/256
Wang et al. [17]	10/256
Özkaynak et al. [10]	10/256
Bhattacharya et al. [24]	10/256

Table 3. A Comparison of LP

S-box	Maximum LP
Proposed S-box	0.1094
Asim et al.[22]	0.1523
Hussain et al. [23]	0.1151
Wang et al. [17]	0.1406
Özkaynak et al. [10]	0.1289
Bhattacharya et al. [24]	0.1328

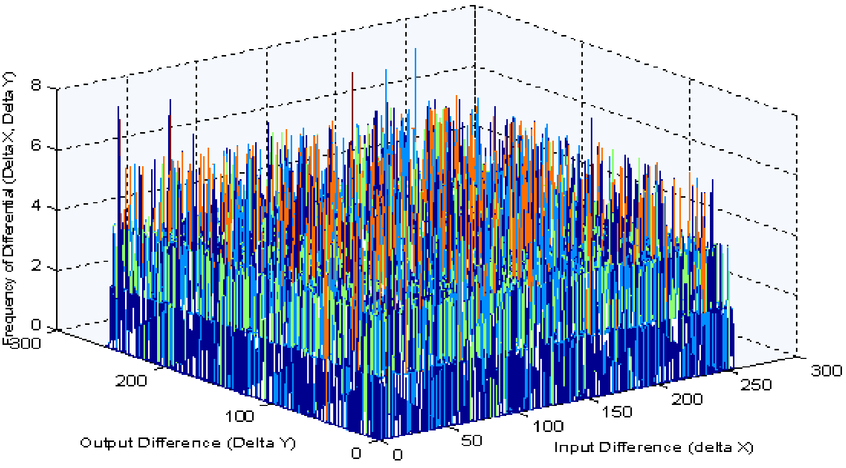


Fig. 1. Difference distribution table of proposed S-box (x-axis input XOR's, y-axis output XOR's, z-axis number of occurrences)

5 Conclusion

This work reports an alternative novel approach to design S-box. Previously, chaos based designed S-box or otherwise, suffers high differential probability. It helps in unveiling the structure of S-box. To overcome the design deficiency and to improve the susceptibility of S-box design against these attacks, it needs to be design systematically. In this work, DDT is used to systematically design and optimize S-box. DDT based S-box design is never reported before. Previously, linear and differential approximation tables are generated to estimate S-box resistance against linear and differential cryptanalysis. Our proposed S-box design achieves low differential probabilities as compared to recently proposed S-box's. Thus, proposed S-box showed more secured against differential and linear cryptanalysis as compared to recently proposed S-box.

References

1. Shannon, C.: Communication theory of secrecy systems. Bell System Technical Journal 28, 656–715 (1949)
2. Amigó, J.M., Kocarev, L., Szczepanski, J.: Theory and practice of chaotic cryptography. Physics Letters A 366, 211–216 (2007)
3. Kocarev, L.: Chaos-based cryptography: a brief overview, Circuits and Systems. IEEE Magazine 1, 6–21 (2001)
4. Kocarev, L., Jakimoski, G.: Logistic map as a block encryption algorithm. Physics Letters A 289, 199–206 (2001)
5. Masuda, N., Aihara, K.: Cryptosystems with discretized chaotic maps. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 49, 28–40 (2002)

6. Szczepanski, J., Amigo, J.M., Michalek, T., Kocarev, L.: Cryptographically secure substitutions based on the approximation of mixing maps. IEEE Transactions on Circuits and Systems I: Regular Papers 52, 443–453 (2005)
7. Behnia, S., Akhshani, A., Ahadpour, S., Mahmodi, H., Akhavan, A.: A. A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps. Physics Letters A 366, 391–396 (2007)
8. Chen, G.: A novel heuristic method for obtaining S-boxes. Chaos, Solitons & Fractals 36, 1028–1036 (2008)
9. Chen, G., Chen, Y., Liao, X.: An extended method for obtaining S-boxes based on three-dimensional chaotic Baker maps. Chaos, Solitons & Fractals 31, 571–579 (2007)
10. Özkaynak, F., Özer, A.B.: A method for designing strong S-Boxes based on chaotic Lorenz system. Physics Letters A 374, 3733–3738 (2010)
11. Tang, G., Liao, X.: A method for designing dynamical S-boxes based on discretized chaotic map. Chaos, Solitons & Fractals 23, 1901–1909 (2005)
12. Tang, G., Liao, X., Chen, Y.: A novel method for designing S-boxes based on chaotic maps. Chaos, Solitons & Fractals 23, 413–419 (2005)
13. Clark, J.A., Jacob, J.L., Stepney, S.: The design of S-boxes by simulated annealing. New Gen. Comput. 23, 219–231 (2005)
14. Fuller, J., Millan, W., Dawson, E.: Multi-objective optimisation of bijective s-boxes. New Generation Computing 23, 201–218 (2005)
15. Laskari, E.C., Meletiou, G.C., Vrahatis, M.N.: Utilizing Evolutionary Computation Methods for the Design of S-Boxes. In: 2006 International Conference on Computational Intelligence and Security, pp. 1299–1302 (2006)
16. Millan, W.L.: How to Improve the Nonlinearity of Bijective S-boxes. In: Boyd, C., Dawson, E. (eds.) ACISP 1998. LNCS, vol. 1438, pp. 181–192. Springer, Heidelberg (1998)
17. Wang, Y., Wong, K.-W., Li, C., Li, Y.: A novel method to design S-box based on chaotic map and genetic algorithm. Physics Letters A 376, 827–833 (2012)
18. Biham, E., Shamir, A.: Differential Cryptanalysis of DES-like Cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1991)
19. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)
20. Schneier, B., Sutherland, P.: Applied cryptography: protocols, algorithms, and source code in C. John Wiley & Sons, Inc. (1995)
21. Bafghi, A.G., Safabakhsh, R., Sadeghiyan, B.: Finding the differential characteristics of block ciphers with neural networks. Information Sciences 178, 3118–3132 (2008)
22. Asim, M., Jeoti, V.: Efficient and simple method for designing chaotic S-boxes. ETRI Journal 30, 170 (2008)
23. Hussain, I., Shah, T., Mahmood, H., Gondal, M.: Construction of S8 Liu J S-boxes and their applications. Computers & Mathematics with Applications 64, 2450–2458 (2012)
24. Bhattacharya, D., Bansal, N., Banerjee, A., RoyChowdhury, D.: A Near Optimal S-Box Design. In: McDaniel, P., Gupta, S.K. (eds.) ICISS 2007. LNCS, vol. 4812, pp. 77–90. Springer, Heidelberg (2007)

On the Extensions of (*k*, *n*)*-Visual Cryptographic Schemes

Kanakkath Praveen, K. Rajeev, and M. Sethumadhavan

TIFAC CORE in Cyber Security
Amrita Vishwa Vidyapeetham
Ettimadai, Amrita Nagar (P.O.)
Coimbatore, India
{praveen.cys,rajeev.cys}@gmail.com, m_sethu@cb.amrita.edu

Abstract. A deterministic (k, n)*-Visual cryptographic scheme (VCS) was proposed by Arumugam et.al [1] in 2012. Probabilistic schemes are used in visual cryptography to reduce the pixel expansion. In this paper, we have shown that the contrast of probabilistic (k, n)*-VCS is same as that of deterministic (k, n)*- VCS. This paper also proposes a construction of (k, n)*-VCS with multiple essential participants. It is shown that in both deterministic and probabilistic cases the contrast of the (k, n)*-VCS with multiple essential participant is same as that of (k, n)*-VCS.

Keywords: Visual Cryptography, Deterministic schemes, Probabilistic schemes.

1 Introduction

Naor and Adi Shamir in 1995 developed a (k, n) OR based deterministic VCS [6] for sharing binary secret images. As an extension of Naor's scheme a deterministic general access structure VCS was introduced by Ateniese et.al. [2] in 1996. Droste [4] in 1998 proposed a (k, n)-VCS with less pixel expansion than Noar's scheme. In 2005 Tuyls et.al invented a XOR based deterministic VCS [7]. The basic parameters for determining the quality of VCS are pixel expansion and contrast. The pixel expansion is a measure of number of sub pixels used for encoding a pixel of secret image while contrast is the difference in grey level between black pixel and white pixel in the reconstructed image. The following are the definitions and notations.

Let $P = \{P_1, P_2, P_3,\ldots, P_n\}$ be the set of participants, and 2^P denote the power set of P. Let us denote Γ_{Qual} as qualified set and Γ_{Forb} as forbidden set. Let $\Gamma_{Qual} \in 2^P$ and $\Gamma_{Forb} \in 2^P$ where $\Gamma_{Qual} \cap \Gamma_{Forb} = \emptyset$. Any set $A \in \Gamma_{Qual}$ can recover the secret image whereas any set $A \in \Gamma_{Forb}$ cannot leak out any secret information. Let $\Gamma_0 = \{A\in \Gamma_{Qual}$: $A' \notin \Gamma_{Qual}$ for all $A' \subseteq A, A' \neq A\}$ be the set of minimal qualified subset of P. The pair $\Gamma = (\Gamma_{Qual}, \Gamma_{Forb})$ is called the access structure of the scheme. Let S be an $n \times m$ Boolean matrix and $A \subseteq P$, the vector obtained by applying the Boolean OR operation to the rows of S corresponding to the elements in A is denoted by S_A. Let $w(S_A)$ denotes the Hamming weight of vector S_A. Let W^0 (resp. W^1) is a set consist of

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 231–238, 2014.

OR-ed value of any k tuple out of n tuple column vector V^0 from S^0 (resp. V^1 from S^1). Let X^0 be a set of values consist of OR-ing 1 with all elements of W^1 and 0 with all elements of W^0. Let X^1 be a set of values consist of OR-ing 1 with all elements of W^0 and 0 with all elements of W^1.

Definition 1 [2]. Let $\Gamma = (\Gamma_{Qual}, \Gamma_{Forb})$ be an access structure on a set of n participants. Two collections of $n \times m$ Boolean matrices S^0 and S^1 constitute a $(\Gamma_{Qual}, \Gamma_{Forb}, m)$ VCS if there exist a positive real number α and the set of thresholds $\{t_A \mid A \in \Gamma_{Qual}\}$ satisfying the two conditions:

1. Any qualified set $A= \{i_1, i_2,\ldots,i_p\} \in \Gamma_{Qual}$ can recover the shared image by stacking their transparencies. Formally $w\ (S^0{}_A) \leq t_A - \alpha.m$, whereas $w(S^1{}_A) \geq t_A$.

2. Any forbidden set $A= \{i_1, i_2,\ldots, i_p\} \in \Gamma_{Forb}$ has no information on the shared image. Formally the two collections of $p \times m$ matrices D_t ,$t \in \{0,1\}$, obtained by restricting each $n \times m$ matrix in S^t to rows $i_1,i_2,\ldots,i_p$ are indistinguishable in the sense that they contain the same matrices with same frequencies. The first property is related to the contrast $\alpha.m$ of the image. The number α is called relative contrast and m is called the pixel expansion. The second property is for the security of the scheme.

Ito et.al [5] in 1999 proposed an image size invariant VCS. In 2003 Yang [9] proposed a non expandable VCS using probabilistic method with same contrast level of expandable VCS. In 2012 Wu et.al [8] proposed a probabilistic VCS for general access structure using random grids. Arumugan et.al [1] in 2012 constructed a deterministic (k, n)*- VCS which is better in pixel expansion than Ateniese scheme. In this paper it is shown that the contrast of probabilistic (k, n)*- VCS is same as that of deterministic (k, n)*- VCS by using Yang's construction. This paper also proposes a construction of (k, n)*- VCS with multiple essential participant. It is shown that in both deterministic and probabilistic cases the contrast of (k, n)*- VCS with multiple essential participant is same as that of (k, n)*- VCS.

2 Preliminaries

2.1 Yang's Construction of Probabilistic (*k*, *n*)- VCS

Let S^0 and S^1 be the basis matrices of a (k, n)-VCS which is of order $n \times m$. When stacking any k rows in S^0 we have m-l white and l black sub pixels. When stacking any k rows in S^1 we have m-h white and h black sub pixels. The value h is defined as the lower bound of the blackness level for encrypting a black pixel in the recovered secret image and l is defined as the upper bound of the blackness level for encrypting a white pixel in the recovered secret image. For sharing a pixel 0 select any one of the column which is of order $n \times 1$ from S^0 and distribute i^{th} row to i^{th} participant and for sharing a pixel 1 select any one of the column which is of order $n \times 1$ from S^1 distribute i^{th} row to i^{th} participant. The appearance probabilities of black color in black (Pr (b/b)) and white areas (Pr (b/w)) are h/m and l/m respectively. Contrast of this scheme is given by $\alpha = (h\text{-}l)/m$.

Example: For a Shamir's (3, 4) - VCS with white and black matrices

$$S^0=\begin{bmatrix}0&0&1&1&1&0\\0&0&1&1&0&1\\0&0&1&0&1&1\\0&0&0&1&1&1\end{bmatrix}\quad S^1=\begin{bmatrix}1&1&0&0&0&1\\1&1&0&0&1&0\\1&1&0&1&0&0\\1&1&1&0&0&0\end{bmatrix}$$

Represent the column vectors of matrix S^0 and S^1 separately as set Y^0= { $\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$, $\begin{bmatrix}0\\0\\0\\0\end{bmatrix}$, $\begin{bmatrix}1\\1\\1\\0\end{bmatrix}$, $\begin{bmatrix}1\\1\\0\\1\end{bmatrix}$, $\begin{bmatrix}1\\0\\1\\1\end{bmatrix}$, $\begin{bmatrix}0\\1\\1\\1\end{bmatrix}$ } and set Y^1= { $\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$, $\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$, $\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$, $\begin{bmatrix}0\\0\\1\\0\end{bmatrix}$, $\begin{bmatrix}0\\1\\0\\0\end{bmatrix}$, $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}$ }.While any 3 participants combines the possible column vectors which generate black pixel(1) are $\begin{bmatrix}0\\1\\1\end{bmatrix}$, $\begin{bmatrix}1\\0\\1\end{bmatrix}$, $\begin{bmatrix}1\\1\\0\end{bmatrix}$, $\begin{bmatrix}1\\0\\0\end{bmatrix}$, $\begin{bmatrix}0\\1\\0\end{bmatrix}$, $\begin{bmatrix}0\\0\\1\end{bmatrix}$, $\begin{bmatrix}1\\1\\1\end{bmatrix}$ and white pixel(0) is $\begin{bmatrix}0\\0\\0\end{bmatrix}$. Then W^0={0,0,1,1,1,1}, W^1={1,1,0,1,1,1},

Pr (*b*/*b*) = 5/6, Pr (*b*/*w*) = 4/6 , which implies α =1/6.

2.2 Construction of Deterministic (*k*, *n*)*-VCS

Let P= {P_1, P_2, P_3, P_4,...., P_n} be the set of participants and SI is the secret binary image to be shared. The minimal qualified sets which will reconstruct SI is defined as Γ_0= {A: $A \subseteq P$, $P_1 \in A$ and $|A|$ =k}. Let S^0 and S^1 be the basis matrices of a (k-1, n-1)-VCS which is of order $n \times m$. When stacking any (k-1) rows in S^0 we have m-l white and l black sub pixels. When stacking any (k-1) rows in S^1 we have m-h white and h black sub pixels. Let $T^0=\begin{bmatrix}0_m 1_m\\ S^0 S^1\end{bmatrix}$ and $T^1=\begin{bmatrix}0_m 1_m\\ S^1 S^0\end{bmatrix}$ are the basis matrices of a (k, n)*- VCS proposed by Arumugam et.al [1].The first row of both T^0 and T^1 consist of m consecutive zeros and m consecutive ones, which corresponds to participant P_1. For i >1, the i^{th} row of T^0 (resp.T^1) is obtained by concatenating the $(i\text{-}1)^{th}$ row of the matrices S^0 (resp. S^1) and S^1 (resp. S^0) which corresponds to remaining participants P_i.

Theorem 1 [1]. Let $\Gamma = (\Gamma_{Qual}, \Gamma_{Forb})$ be any access structure on $R=\{P_2, P_3,\ldots,P_n\}$.Let $\Gamma^{*} = (\Gamma_{Qual}{}^{*}, \Gamma_{Forb}{}^{*})$ be the corresponding access structure on $P=\{P_1, P_2, P_3,\ldots,P_n\}$. Given a VCS for the access structure Γ with pixel expansion m and relative contrast α, then there exist a VCS for Γ^{*} with pixel expansion $2m$ and relative contrast $\alpha/2$.

3 Probabilistic (*k*, *n*)*-VCS Using Yang's Scheme

Let T^0 and T^1 be the basis matrices for a deterministic $(k, n)^{*}$-VCS. The sharing of white pixel is done by randomly selecting one column V^0 from T^0 and sharing of black pixel is done by randomly selecting one column V^1 from T^1 respectively. The chosen column vector $V^{g} = [v_u]$ ($1 \le u \le n$, $g \in \{0, 1\}$) defines the color of the pixel in the shared images. Each v_u is interpreted as black if $v_u=1$ and white if $v_u=0$. Let the size of SI is given as $(p \times q)$. The share Sh_u, $1 \le u \le n$ for the participant P_u is $Sh_u(m,h) = \{v_u \ , \ 1 \le u \le n\}$, $1 \le m \le p$, $1 \le h \le q$. For proof of security, let $A \notin \Gamma_{Qual}$ be the participant set then $w(T^0{}_A) = w(T^1{}_A)$ implies contrast $\alpha'=0$. For $A \in \Gamma_{Qual}$ then $w(T^0{}_A) < w(T^1{}_A)$ then Pr (b/b) in $T^1{}_A$ is $(m + h)/2m$ and that in $T^0{}_A$ is $(m + l)/2m$ .So contrast $\alpha' = ((m + h)/2m) - ((m + l)/2m) = (h-l)/2m = \alpha/2$.

The same construction can be done using visual secret sharing for general access structures using random grids introduced by Wu et.al [8]. But contrast of the proposed probabilistic method of $(k, n)^{*}$-VCS is completely determined by the basis matrices S^0 and S^1. Different basis matrices would lead to different contrast. So the comparison of proposed scheme with Wu's [8] scheme is not possible.

Example: Let the binary secret image need to be shared for (3, 4)*-VCS is SI= $\begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ .Let us represent $S^0 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ and $S^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ where $\alpha = 1/3$. The basis matrices T^0 and T^1 constructed using Arumugam scheme for the participant set $\{P_1, P_2, P_3, P_4\}$ is given by $T^0 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ $T^1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$.

The minimal qualifies sets for reconstructing SI is $\Gamma_0 = \{\{P_1, P_2, P_3\}, \{P_1, P_2, P_4\}, \{P_1, P_3, P_4\}\}$. The forbidden set is given by $\Gamma_{Forb} = \{ P_1, \{ P_2, P_3\},\{ P_2, P_4\},\{ P_3, P_4\}, \{ P_2, P_3, P_4\},\{ P_1, P_3\},\{ P_1, P_2\},\{ P_1, P_4\}\}$. The elements of the set Γ_{Forb} will not leak out any information about SI. The shares Sh_i of the participants for probabilistic (3, 4)*-VCS are constructed by selecting one column from T^0 (resp.T^1).The shares are

$$Sh_1=\begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, Sh_2=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, Sh_3=\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

$Sh_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$. Then W^0={1,0,0}, W^1={1,1,0}, X^0={1,1,1,1,0,0} and X^1={1,1,1,1,1,0}.The values of *h*, *l* and *m* are 2,1 and 3 respectively. The probability of occurrence of 1(black) in X^0(resp. X^1) is Pr (*b*/*w*) = 4/6 (resp. Pr (*b*/*b*) = 5/6), which implies α'=1/6.

4 Probabilistic (*k*, *n*)*-VCS with Perfect Reconstruction of Black Pixel Using Yang's Scheme

In 2001 Blundo et.al [3] proposed a VCS in which the reconstruction of black pixel is perfect. Let S^0 and S^1 be the basis matrices of a (*k*-1, *n*-1)-VCS which will reconstruct the black pixel perfect. In this construction the reconstruction of black pixel is perfect but the reconstruction of white pixel is probabilistic. The basis matrices T^0 (resp.T^1) is constructed using S^0 and S^1. The share construction of this scheme is same as that of probabilistic (*k*, *n*)*-VCS. For proof of security, let $A \notin \Gamma_{Qual}$ be the participant set then $w(T^0{}_A) = w(T^1{}_A)$ implies α =0 .For $A \in \Gamma_{Qual}$ then $w(T^0{}_A) < w(T^1{}_A)$ then probability of blackness in $T^1{}_A$ is $2m/2m$ and that in $T^0{}_A$ is $(m + l)/2m$ .This implies that reconstruction of black pixel is perfect but the reconstruction of white pixel is probabilistic. So according to Ito's definition contrast $\alpha'= (2m/2m) - ((m + l)/2m) = (m\text{-}l)/2m= \alpha/2$.

Example: Let the basis matrices for black and white pixel for a (3, 6)-VCS with perfect reconstruction of black and white pixel is S^0 and S^1.The basis matrices for the construction of (4, 7)*-VCS are $T^0=\begin{bmatrix} 0_{25}1_{25} \\ S^0S^1 \end{bmatrix}$ and $T^1=\begin{bmatrix} 0_{25}1_{25} \\ S^1S^0 \end{bmatrix}$.Let the matrices S^0 and S^1 is represented as

$$S^0=\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$S^1=\begin{bmatrix} 0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 0&1&1&1&1&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 1&0&1&1&1&0&1&1&1&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1\\ 1&1&0&1&1&1&0&1&1&0&1&1&0&0&1&1&1&1&1&1&1&1&1&1&1\\ 1&1&1&0&1&1&1&0&1&1&0&1&0&1&0&1&1&1&1&1&1&1&1&1&1\\ 1&1&1&1&0&1&1&1&0&1&1&0&1&0&0&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$$

where α =1/25.Then W^0={0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
W^1={1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
X^0={0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1},
X^1={1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1}. The probability of occurrence of 1(black) in X^0(resp. X^1) is Pr (b/w) = 49/50 (resp. Pr (b/b) = 1) which implies that contrast α' =1/50.

5 Construction of (k, n)*-VCS with Multiple Essential Participants

Let $P = \{P_{1,} P_2, P_{3,}\dots, P_n\}$ be the set of participants and SI be the secret image to be shared. Let Γ_{Qual} be the family of qualified sets for SI and Γ_{Forb} be the family of forbidden sets. The minimal qualified sets which will reconstruct SI are defined as Γ_{E0*}= $\{B\text{: } B \subseteq P, \{P_1, P_2, P_{3\dots} P_t\} \in B$ and $|B|$ = $k\}$. Let S^0 and S^1 be the basis matrices of a (k-t, n-t) -VCS. The construction of deterministic (k, n)*-VCS is done as follows. Let C_t^{00} and C_t^{11} be the matrices of size (t×m). Let us define ev (resp. od) as even (resp. odd) number. The matrix C_t^{00} contains $(ev)1m$ row vectors and $(t\text{-}ev)0m$ row vectors. The matrix C_t^{11} contains (od) $1m$ row vectors and $(t\text{- } od)$ $0m$ row vectors. Let us define two matrices $T^0=\begin{bmatrix} C_t^{00} & C_t^{11} \\ S^0 & S^1 \end{bmatrix}$ and $T^1=\begin{bmatrix} C_t^{00} & C_t^{11} \\ S^1 & S^0 \end{bmatrix}$.Any possible column permutation of C_t^{11} and C_t^{00} can be used in T^0 and T^1. The first t rows of both T^0 and $T^1$is obtained by concatenating the i^{th} row of the matrices C_t^{00} and C_t^{11}. For $i > t$, the i^{th} row of T^0 (resp.T^1) is obtained by concatenating the $(i\text{-}t)^{th}$ row of the matrices S^0 (resp. S^1) and S^1 (resp. S^0). The first i^{th} row of both T^0 and T^1 corresponds to shares Sh_i of participants P_i. Let if S^0 and S^1 be the basis matrices of a (k-t, n-t)-VCS which will reconstruct the black pixel perfect. Then the basis matrices T^0 and T^1 can be used for constructing a (k, n)*-VCS with multiple essential participants which reconstructs the black pixel perfect.

In the construction of probabilistic (k, n)*-VCS with multiple essential participants the sharing of white pixel is done by randomly selecting one column V^0 from T^0 and sharing of black pixel is done by randomly selecting one column V^1 from T^1 respectively. The chosen column vector $V^g = [v_u]$ ($1 \le u \le n$, $g \in \{0, 1\}$) defines the color of the pixel in the shared images. Each v_u is interpreted as black if v_u=1 and white if

v_u=0. Let the size of SI is given as $(p \times q)$.The shares Sh_u, $1 \le u \le n$ for the participants P_u are generated as $Sh_u(m,h) = \{v_u \quad , \quad 1 \le u \le n\}$; where $1 \le m \le p$, $1 \le h \le q$. During decryption OR-ing any *k*-*t* of the *n*-*t* shares of Sh_u, where $(t+1) \le u \le n$ and then OR the result with XOR-ed output of all Sh_u, $1 \le u \le t$ to reconstruct SI. For both probabilistic and deterministic (*k*, *n*)*-VCS with multiple essential participants contrast is same as that of (*k*, *n*)*-VCS.

Example: Let $C_3^{00} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ and $C_3^{11} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$. Let us represent $S^0 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ and $S^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ where α =1/3.The basis matrices T^0 and T^1 for the participant set $\{P_1, P_2, P_3, P_4, P_5, P_6\}$ for Probabilistic (5, 6)*-VCS with multiple essential participants is given by

$$T^0 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad T^1 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

. The minimal qualified set which can reconstruct SI is $\Gamma_{E0^*} = \{\{P_1, P_2, P_3, P_4, P_5\}, \{P_1, P_2, P_3, P_5, P_6\}, \{P_1, P_2, P_3, P_4, P_6\}\}$. Represent the column vectors of matrix T^0 and T^1 separately as set

$$Y^0 = \{ \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \} \text{ and } Y^1 = \{ \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \}$$

.While any 5 participants combines (3 of them are essential participants) the possible column vectors which generate black(1) pixel are $\begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ and white pixel(0) is $\begin{bmatrix}1\\0\\1\\0\\0\end{bmatrix}$. Then X^0={1,0,0,1,1,1}, X^1={1,1,0,1,1,1}, Pr (*b*/*b*) = 5/6, Pr (*b*/*w*) = 4/6 , which implies α'=1/6.

6 Conclusion

In this paper it is shown that the contrast of deterministic (*k*, *n*)*- VCS is same as that of probabilistic (*k*, *n*)*-VCS. We also proposes a (*k*, *n*)*-VCS with multiple essential participants where the contrast is same as that of (*k*, *n*)*- VCS.

References

1. Arumugam, S., Lakshmanan, R., Nagar, A.K.: On *(k, n)**-Visual Cryptography Scheme. Designs, Codes and Cryptography (2012), doi:10.1007/s10623-012-9722-2
2. Ateniese, G., Blundo, C., De Santis, A., Stinson, D.R.: Visual Cryptography for General Access Structures. Information and Computation 129(2), 86–106 (1996)
3. Blundo, C., Bonis, A.D., De Santis, A.: Improved schemes for visual cryptography. Designs, Codes and Cryptography 24(3), 255–278 (2001)
4. Droste, S.: New Results on Visual Cryptography. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 401–415. Springer, Heidelberg (1996)
5. Ito, R., Kuwakado, H., Tanaka, H.: Image size invariant visual cryptography. IEICE Trans. Fundamentals E82-A(10), 2172–2177 (1999)
6. Naor, M., Shamir, A.: Visual cryptography. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 1–12. Springer, Heidelberg (1995)
7. Tuyls, P., Hollmann, H.D.L., Lint, J.H.V., Tolhuizen, L.: XOR-based visual cryptography schemes. Designs, Codes and Cryptography 37(1), 169–186 (2005)
8. Wu, X., Wei, S.: Visual secret sharing for general access structures by random grids. Information Security, IET 4(6), 299–309 (2012)
9. Yang, C.N.: New Visual Secret Sharing Schemes Using Probabilistic Method. Pattern Recognition Letters 25(4), 481–494 (2004)

A Graph Data Model for Attack Graph Generation and Analysis

Mridul Sankar Barik and Chandan Mazumdar

Dept. of Comp. Sc. and Engg., Jadavpur University Kolkata, India
{msbarik,chandanm}@cse.jdvu.ac.in

Abstract. Attack graph is a useful tool for enumerating multi-stage, multi-host attacks in organizational networks. It helps in understanding the diverse nature of threats and to decide on countermeasures which require on-the-fly implementation of custom algorithms for attack graph analysis. Existing approaches on interactive analysis of attack graph use relational database which lack data structures and operations related to graph. Graph databases enable storage of graph data and efficient querying of such data. In this paper, we present a graph data model for representing input information for attack graph generation. Also, we show how graph queries can be used to generate attack graph and facilitate its analysis.

Keywords: Attack Graph, Graph Database, Graph Query.

1 Introduction

With ICT based systems becoming ever pervasive, securing such systems poses a great challenge. Number and types of attacks against such systems are on the rise and is a growing concern for security administrators. Incidentally, many of these attacks when considered in isolation are very simple and easy to detect. But in most of the cases misfeasors combine those attacks to launch multistage attacks against critical assets. Conventional defense approaches have been mostly host centric, where attention is given on identifying vulnerabilities of the individual hosts and taking measures to mitigate them. But these methods are less effective on the face of multistage attacks.

Attack graph has been a useful tool for enumerating multi-stage, multi-host attacks in organizational networks. Without this tool it is very difficult even for experienced security analysts to manually discover, how an attacker can combine vulnerabilities in the same host or in connected hosts, to compromise critical resources in a manner hitherto unknown. And the task becomes even more difficult as the number of different vulnerabilities as well as the size of the network increase. An attack graph that shows all possible multi-stage, multi-host attack paths, is crucial to a security administrator, as it helps in understanding the diverse nature of threats and to decide on appropriate countermeasures. All this call for efficient scalable solution for attack graph generation and its analysis. Prior research work in this area tried to address these issues.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 239–250, 2014.

Most of the existing approaches of attack graph analysis use proprietary algorithms and do not necessarily always use standard graph based algorithms only. In many situations, such analysis techniques may need to be adapted frequently due to changes in host/network configuration and/or security objectives. Thus, there is always a need for a solution which allows interactive analysis of attack graph. Wang et al. [14] first addressed this issue and proposed a relational model for representing network configurations and domain knowledge. They showed how attack graph can be generated from those inputs as relational views and also typical attack graph analysis operations to be realized as relational queries. The objective of such a solution was to enable security analysts to implement custom analysis algorithms on-the-fly in terms of relational queries, thereby reducing delay in providing responses to dynamic network conditions. But, it is now a well accepted fact that for exploration of large graph data such as attack graphs, relational database is not the most practical and scalable solution. This is mainly due to lack of data structures and operations related to graphs in relational databases. In recent years there has been a re-emergence of interests in graph databases for storing and managing large graph data such as social graphs, web graphs, biological graphs etc. Use of graph databases in comparison to relational databases, immensely increases performance when dealing with connected data.

In this paper, we present a graph data model for representing input information for generation of attack graph. Further, we show how graph queries can be used to generate attack graph and perform typical analysis tasks over the generated attack graph. We have used the popular Neo4j graph database for implementing the proposed model. The remainder of this paper is structured as follows. Section 2 surveys related work on attack graphs and graph databases. Section 3 describes our proposed graph data model and section 4 sketches the attack graph generation methodology. Section 5 concludes the paper.

2 Related Works

2.1 Attack Graphs

Early research on attack graphs concentrated on its efficient generation. Some of the initial approaches used symbolic model checking [9][10] to analyze the model of individual host configuration and vulnerabilities. The security requirements/properties of the system are specified as logic statements. The counterexamples produced by the model checker are used to build the attack graph. But these approaches suffered from scalability issues as the number of nodes, representing possible system states explodes with the increase of network hosts. Automated approaches based on custom algorithms tried to address the scalability issues by considering each node of the attack graph to represent different network configurations rather than individual system states. The TVA approach [4] is the most notable among them. It assumes the monotonicity property of attacks which guarantees that no action of an attacker can interfere with the attacker's ability to take any other action. The TVA approach follows an exploit dependency graph to represent the pre and post conditions of an exploit. It then uses a graph search algorithm to chain the individual vulnerabilities and find attack paths comprising multiple vulnerabilities.

The NetSPA tool [6] generates attack graph from firewall rule sets and network vulnerability scans. It then produces a set of prioritized recommendations by analyzing the attack graph. MULVAL (Multihost, multistage, Vulnerability Analysis) [8] is an attack graph generation tool which uses Datalog as its modeling language. The authors introduced the notion of logical attack graph which specifies the causality relationships between system configuration information and an attacker's potential privileges, expressible in the form of a propositional formula. The reasoning engine captures, through automatic logic deductions, the interaction among various components in the network which eventually leads to the formation of the attack graph. Chen et al. [2] introduced an attack pattern oriented method for full attack graph generation. The authors considered an attack pattern as an abstract description of the common approach attackers take to exploit analogous vulnerabilities. They have developed a set of attack patterns, and mapped vulnerabilities to corresponding attack patterns. However, their approach uses a custom algorithm for attack graph generation.

Attack graphs have been used for measuring network security or as an aid to many security solutions. Intrusion alert correlation techniques [12] try to detect multi-step intrusions by correlating isolated alerts of individual attack steps. This method first maps a currently received intrusion alert to the corresponding exploit in the attack graph. Then, it reasons about previous exploits or alerts that prepare for the current one and possible exploits in the future. Network hardening solution proposed in [13] uses attack paths enumerated in an attack graph. It provides a solution which identifies vulnerabilities that should be removed such that none of the attack paths leading to a given critical resource can be realized and also the cost of removing such vulnerabilities is least. Idika and Bhargava [3] proposed a suite of security metrics based on characteristics of different attack paths in an attack graph. In most of the cases, security analysis techniques based on attack graphs concentrate on properties of different attack paths in them.

Most of the existing works on attack graph generation and analysis are silent about issues regarding efficient storage and retrieval of attack graphs, such that the subsequent analysis tasks become easier and can be done on the fly. This fact has motivated our present work, to use graph database for generation, storage and query of attack graphs to enhance existing analysis techniques and also to open up opportunities for new types.

2.2 Graph Databases

Recently, there has been a re-emergence of interests in graph databases for storing and managing large graph structured data such as social graphs, web graphs, biological graphs etc. Existing relational data modeling follows strict data model and are not suitable for semi structured data. Moreover, SQL, the standard query language for relational databases cannot express paths which are essential for graph based reasoning. On the other hand, graph databases use powerful data model suitable for semi structured and connected data. Also the graph query languages have been specifically designed for graph based reasoning.

A graph database exposes a graph data model and has support for basic Create, Read, Update, and Delete (CRUD) methods. In comparison to the index-intensive, set theoretic operations of relational databases, graph databases make use of index free traversals. Most of the graph databases use native graph storage, which is optimized for storing and managing graphs and native graph processing engines that leverage index free adjacency. In such graph databases, called the native graph databases, relationships attached to a node naturally provide a direct connection to other related nodes of interest. Graph queries as a result, mostly involve using this locality to traverse through the graph, in contrast to joining data through a global index, which are many orders of magnitude slower.

For attack graph analysis, graph database is suitable as the analysis task in most of the cases involves local processing around a node within a larger global attack graph. Very few analysis are based on global characteristics of attack graph.

In our implementation of the proposed graph data model we have used the Neo4j [7] graph database. It has native processing capabilities as well as native graph storage. Neo4j has a built in graph query language Cypher. It enables a user to ask the database to find data that matches a specific pattern.

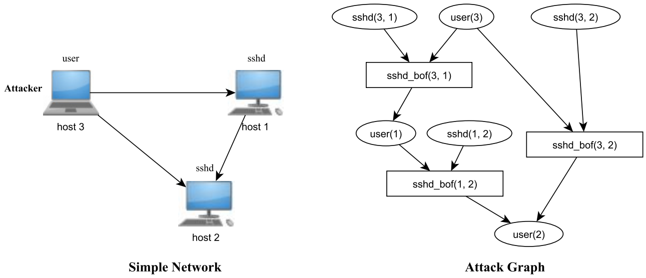
Till now, there have been limited cases of graph databases being used in cyber security solutions. In a recent work, Joslyn et al. [5] have presented a query language for IPFLOW data. The authors have shown that many widely known cyber attack scenarios can be expressed as graph pattern queries over IPFLOW data when treated as a large graph. They have presented such queries in SAPRQL and Datalog graph query languages for two kinds of attack scenarios, exfiltration and hierarchical botnet. In another work Chen et al.[1] have presented a high level abstract query language for detection of coordinated scanning activity from attack data collected from firewalls, intrusion detection systems and honeynets. This work however uses a relational database as backend data store. The abstract query language presented in this paper, allows users to express specific analysis tasks and a query processor translates them into SQL queries for the backend relational database.

3 Proposed Model

3.1 Attack Graph Semantics

A number of formalisms have been reported in literature for attack graph representation. We have used the exploit dependency graph representation as given in [4]. Formally, an attack graph is represented as a directed graph $G = (V, E)$. Set of nodes in this graph can be of two types. One type of nodes represent either some attacker capability or some network conditions. Other type of nodes represent exploits. Directed edges from first type of nodes to second type represent prerequisites or preconditions of an exploits. Directed edges from second type of nodes to first type represent effect or postconditions of executing any exploit. This representation is based on the monotonicity assumption, which states that preconditions of a given exploit are never invalidated by the successful application of another exploit.

Attacker's capabilities describe the fact that the attacker has a capability on a specific host. Attacker's capabilities $user(1)/superuser(1)$ says that attacker has $user/superuser$ capability at $host1$. Network conditions describe the fact that either a service running on a destination host can be accessed from a source host or a unidirectional relation that exists from source host to destination host. Network condition $ftp(1, 2)$ means that the ftp service running on $host2$ is accessible from $host1$. Network condition $trust(1,\ 2)$ means that a trust relationship exists from $host1$ to $host2$ allowing any user on $host2$ to remotely execute shell commands on $host1$ without providing any password.



Simple Network

Attack Graph

Fig. 1. An simple network and its corresponding attack graph

The other types of nodes represent exploits. To execute an exploit, an attacker may require multiple capabilities or network conditions known as preconditions of the exploit. Successful execution of an exploit may create new attacker capabilities or new network conditions known as postconditions of the exploit. Exploit $e(1,\ 2)$ describes the fact that an attacker having some capability on $host1$ with the help of some existing network conditions, can perform the exploit e, exploiting some vulnerability on destination $host2$ thereby gaining new capabilities or establishing new network conditions. Exploit $e(1)$ means that the exploit e is performed locally at $host1$. For example, buffer overflow vulnerabilities CVE-2002-0640 of $sshd$, CVE-2003-0466 of $wu - ftpd$, CVE-2003-0245 of $apache$ server etc., allow attacker from a remote machine to execute arbitrary code, thereby possibly gaining user privilege. CVE-2004-0495 is a vulnerability in Linux kernel which allows local user to gain privileges.

Example: Figure 1 shows a simple example network that we shall consider as a running example throughout this paper. The network consists of three hosts $host1$, $host2$, and $host3$. $host1$ and $host2$ run the $sshd$ service and it has a vulnerability CVE-2002-0640. $sshd$ service instance running at $host2$ can be accessed from $host3$ and $host1$ and the other instance of $sshd$ service at $host1$ can be accessed from *host*3 only. The attacker has initially user privilege at *host*3. To the right of the example network is the attack graph of this simple network configuration, where ovals show the network conditions and the rectangles show exploits.

3.2 The Graph Data Model

This section provides a graph data model for representing the input information required for attack graph generation as well as the attack graph itself for facilitating subsequent analysis.

For generating attack graph two types of information i.e. network configuration and domain knowledge are necessary. The network configuration information includes network topology information, firewall (perimeter and/or host based) rule set and per host vulnerability information. The domain knowledge refers to the interdependency between different types of vulnerabilities and network conditions.

In our approach we have modeled network configuration information as graph data and the domain knowledge is encoded as graph patterns. Graph queries are used to look for existence of such patterns over the graph data representing network configuration information. Results of those queries provide information about which vulnerabilities can be exploited based on the present network configuration information. The exploitation of such vulnerabilities may generate new network conditions which we also model as graph patterns i.e. set of new nodes and edges, which are added to the existing graph data.

The conceptual graph data model is shown in Figure 2. The graph model consists of set of nodes V and set of edges E. Nodes can represent either entities V_E or facts V_F. So, we have $V = V_E \cup V_F$. Edges represent relationships between entities or between entities and facts. The basic entities in this model are **hosts** (H), **services** (S), **vulnerabilities** (V), **attacker** (A) and **attacker goals** (G). Each type of entity may have any number of properties as key-value pairs which uniquely describe those entities.

The other type of nodes in the graph data model are the fact nodes which represent interaction among entity nodes. A **service instance** (SI) fact node represents the fact that a specific service S is running at a specific host H. One SI node is connected to a service node via INSTANCE_OF relationship and to a host node via AT_HOST relationship. A **service access instance** (SAI) fact node represents the fact that a specific host H can access a specific service instance SI. One SAI node is connected to a service instance node via ACCESS_TO relation and to a host node via ACCESS_BY relation. A service access instance node enables capturing of information on whether from a source host a service running on a destination host is accessible. This may otherwise be prohibited by a host based firewall running at the destination host. The **host relation** (HR) nodes capture directed relations that hold between any two hosts. One HR node is connected to the source host node of the relation via a FROM relationship and to the destination node via a TO relationship. The **goal instance** (GI) fact nodes represent the fact about any goal type G which the attacker has achieved at a specific host H. One GI node is connected to a goal type node via INSTANCE_OF relationship and to a host node via AT_HOST relationship. The attacker must have some initial privileges in form of some goal instances. The attacker node is connected to such goal instances via START_GOAL relationships. The goal instances together with service access instances and host relations act as preconditions which enable the attacker to launch attacks by exploiting vulnerabilities of accessible services. We call them as **attack instances** (AI).

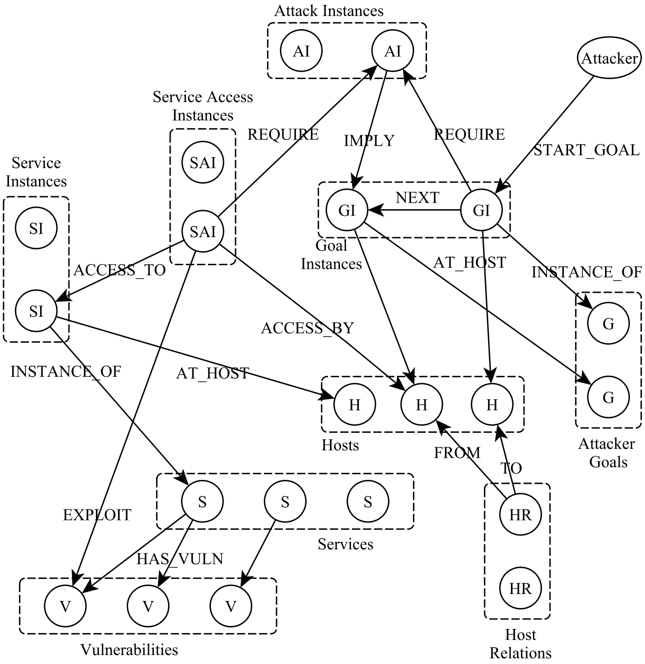


Fig. 2. Graph data model for network configuration information and attack graph

The attack instances fact nodes are equivalent to the exploit nodes in the attack graph semantics discussed in Section 3.1. Whereas, the goal instance facts are equivalent to attacker's capability nodes, the service access instance and host relation facts to network conditions. The preconditions of an attack instance are connected to the attack instance node via REQUIRE relationships. Successful execution of an attack either enables the attacker to gain privileges in form of some attacker goals achieved at some host i.e. new goal instances or to create some new service access instances or host relations. The attack instance node is connected to those new nodes via IMPLY relationships. Also, each goal instance node pair (m, n) which appear as one of the precondition (via REQUIRE relationship) and postcondition (via IMPLY relationship) respectively of an attack instance node, are connected via a directed relationship NEXT from m to n.

This relationship captures the temporal order among goal instances that the attacker achieves in a chain of exploits.

Figure 3 shows the graph data corresponding to the simple example network of our running example. The *sshd* service has a vulnerability CVE-2002-0640 which allows remote attacker to gain user privilege on the host running the service. *sshd*(1) and *sshd*(2) are the service instance fact nodes representing the facts that there are two instances of *sshd* service running at *host*1 and *host*2 respectively. The service access instance fact nodes *sshd*(3, 1), *sshd*(3, 2), *sshd*(1, 2) represent facts about different hosts from which those service instances can be accessed. The goal instance node *user*(3) means that the attacker has user privilege at *host*3 and this is the only privilege the attacker initially has.

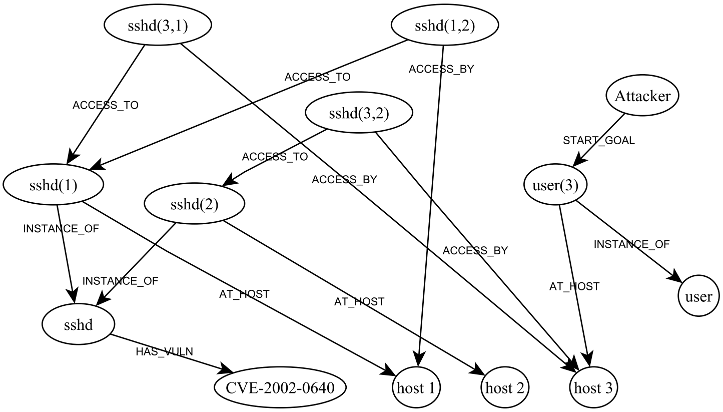


Fig. 3. Graph data of example network

The service access instance nodes uniquely associate a host and a service instance node. This allows a single service instance to be exploited from different hosts. But a service may have multiple vulnerabilities. We need to be able to identify whether through a given service access instance, a specific vulnerability of a service has been exploited or not. For that, once such a vulnerability is exploited a relationship EXPLOIT is added from the services access instance node to the vulnerability node which has been exploited.

4 Generation of Attack Graph

The attack graph generation methodology in our case is an attack pattern based approach. The proposed methodology builds the attack graph in an iterative manner. In each iteration, from the present location of the attacker, all accessible services with vulnerabilities not yet exploited are found out. Then those vulnerabilities are mapped to corresponding attack patterns and the associated actions are performed. The actions associated with an attack pattern involve checking whether all the preconditions for successful exploitation of the associated vulnerability are satisfied or not and if satisfied then create postconditions as applicable. We assume the existence of such a mapping AP which maps a given vulnerability v, to its corresponding attack pattern $AP(v)$. Also, for an attack pattern (ap), *ap.precond* denotes the set of preconditions and *ap.postcond* denotes the set of postconditions. **Algorithm 1** briefly sketches the steps in the attack graph generation methodology.

```
   Input: graph data of present network configuration
   Output: graph data of corresponding attack graph
   begin
1      Find source hosts src where the attacker has user/superuser
       privilege
2      Find all services s, running at destination hosts dst such that s is
       accessible from the source host src
3      Find any vulnerability v of the service s not yet exploited from the
       source host src
4      if no such vulnerability found in step 3 then
           Stop
       else
           foreach vulnerability v of service s do
               patternX(src, dst, s, v)
           end
       end
5      Goto step1
   end
```

Algorithm 1. Attack graph generation

Example queries in Cypher graph query language, which accomplishes the steps 1 to 3 as enumerated above are shown below. We assume that, an attacker goal node has a property *name* which states the kind of goal it is.

```
Step1:  MATCH (src)<-[:AT_HOST]-(gi)-[:INSTANCE_OF]->(g)
        WHERE g.name="user" OR g.name="superuser"
        RETURN src
Step2:  MATCH (src)<-[:ACCESS_BY]-(sai)
        RETURN src, sai
Step3:  MATCH (sai)-[:ACCESS_TO]->(si)-[:INSTANCE_OF]->(s)
        -[:HAS_VULN]->(v), (si)-[:AT_HOST]->(dst), (sai)-[?r]->(v)
        WHERE r is null
        RETURN s, v, dst
```

Each of these queries essentially searches for occurrence of a specific graph pattern in the underlying graph data. When executed on the graph data of our running example, result of these queries show that the attacker from its initial location at *host*3 can access the *sshd* service instances running at *host*1 and *host*2 and both instances have unexploited vulnerability CVE-2002-0640. Next, the attack pattern associated with this vulnerability is instantiated. **Algorithm 2** describes briefly the template of an attack pattern.

```
patternX(src, dst, s, v)
src: source host,
dst: destination host,
s: service,
v: vulnerability
begin
1    ap = AP(v)
2    if all conditions $c \in ap.precond$ are satisfied then
3        Create a new attack instance node nai
4        Create REQUIRE relationships, from all $c \in p.precond$ to nai
5        Create IMPLY relationships from nai to all $c \in ap.postcond$
6        Create NEXT relationships from all $c \in ap.precond$ to all
         $c\prime \in ap.postcond$ such that c and $c\prime$ are goal instance type nodes
7        Create EXPLOIT relationship from the service access instance
         sai node (which allowed attacker at src, access to service s at
         dst) to v
     end
end
```

Algorithm 2. Attack pattern template

Step 2 in above algorithm, translates into pattern queries (similar to the ones shown previously), which determine whether all the preconditions for the current attack pattern are satisfied or not. Step 3-7 simply creates new nodes, relationships and can be easily achieved through Cypher queries. The actions associated with the attack pattern for the vulnerability CVE-2002-0640 of *sshd* service doesn't require any extra activities as the two preconditions necessary for exploiting the vulnerability i.e. *sshd*(*src*, *dst*) and *user*(*src*), are already part of the activity for finding out the accessibility of the vulnerable service itself. The associated attack pattern however creates new nodes and relationships as indicated in the template. Figure 4 shows the full attack graph of our running example.

The generated attack graph is stored in the same graph database, with direct links to input information for each exploit i.e. attack instance (although it is not shown in the figure for clarity in presentation). Our approach is advantageous compared to approaches based on relational databases [14] where large table joins are necessary for determining network conditions that enables an attacker to execute exploits. Another key benefit is that Cypher graph query language has rich support for path queries. Most of the attack graph based analysis tasks discussed in Section 2 can be easily implemented using Cypher path queries. Following is a simple query in Cypher which returns an attack path (if exists)

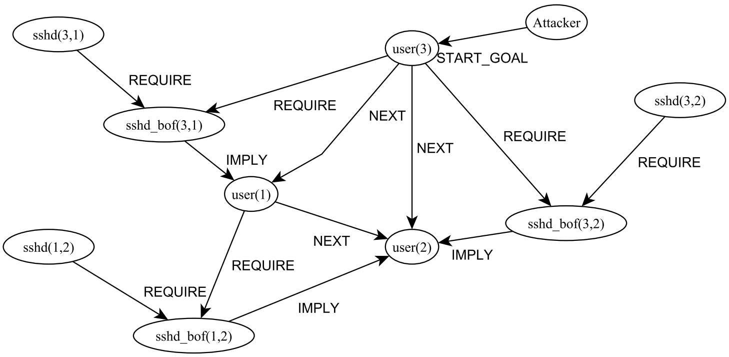


Fig. 4. Graph data of generated attack graph

from a source host to a destination. We assume that, a host node has property *ip_addr* which uniquely identifies it.

```
MATCH (src)<-[:AT_HOST]-(gi1)-[:INSTANCE_OF]->(g),
(dst)<-[:AT_HOST]-(gi2)-[:INSTANCE_OF]->(g)
WHERE g.name="user" AND
src.ip_addr="1.1.1.1" AND dst.ip_addr="2.2.2.2"
WITH gi1, gi2 MATCH
p=(gi1)-[:NEXT*..]->(gi2)
RETURN p
```

5 Conclusion

Attack graphs are useful tool for modeling attacker behavior especially in situations where an attacker may combine vulnerabilities across different hosts to compromise critical resources. Generating attack graphs for typical enterprise networks is a challenging task. In this paper we proposed a graph data model for representing input information for generating attack graph. We have used popular Neo4j graph database for graph information storage. Also we have shown, from this information how attack graph can be generated using simple graph queries in Cypher graph query language. Going by the current trend in graph database research which is fast maturing, well supported by benchmark results reported in literature [11], we envisage the proposed solution as viable. In future work, we plan to augment our model to incorporate different types of network conditions and attacker goals so as to cover wide range of vulnerabilities and attack patterns.

References

1. Chen, B., Yegneswaran, V., Barford, P., Ramakrishnan, R.: Toward a Query Language for Network Attack Data. In: Proceedings of the 22nd International Conference on Data Engineering Workshops, pp. 28–28. IEEE Press, New York (2006)
2. Chen, F., Su, J., Zhang, Y.: A Scalable Approach to Full Attack Graphs Generation. In: Massacci, F., Redwine Jr., S.T., Zannone, N. (eds.) ESSoS 2009. LNCS, vol. 5429, pp. 150–163. Springer, Heidelberg (2009)
3. Idika, N., Bhargava, B.: Extending Attack Graph-Based Security Metrics and Aggregating Their Application. IEEE Transaction on Dependable and Secure Computing 9(1), 75–85 (2012)
4. Jajodia, S., Noel, S., O'Berry, B.: Topological Analysis of Network Attack Vulnerability. In: Kumar, V., Srivastava, J., Lazarevic, A. (eds.) Managing Cyber Threats. LNCS, pp. 247–266. Springer (2005)
5. Joslyn, C., Choudhury, S., Haglin, D., Howe, B., Nickless, B., Olsen, B.: Massive scale cyber traffic analysis: a driver for graph database research. In: First International Workshop on Graph Data Management Experiences and Systems, pp. 3:1–3:6. ACM, New York (2013)
6. Lippmann, R., Ingols, K., Scott, C., Piwowarski, K., Kratkiewicz, K., Artz, M., Cunningham, R.: Validating and restoring defense in depth using attack graphs. In: Proceedings of the IEEE Conference on Military Communications, pp. 981–990. IEEE Press, Piscataway (2006)
7. Neo4j Graph Database, `http://www.neo4j.org/`
8. Ou, X., Boyer, W.F., McQueen, M.A.: A scalable approach to attack graph generation. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 336–345. ACM, New York (2006)
9. Ritchey, R.W., Ammann, P.: Using model checking to analyze network vulnerabilities. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 156–165. IEEE Press, New York (2000)
10. Sheyner, O., Haines, J.W., Jha, S., Lippmann, R., Wing, J.M.: Automated Generation and Analysis of Attack Graphs. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 273–284. IEEE Press, New York (2002)
11. Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., Wilkins, D.: A comparison of a graph database and a relational database: a data provenance perspective. In: Proceedings of the 48th Annual Southeast Regional Conference, pp. 42:1–42:6. ACM, New York (2010)
12. Wang, L., Liu, A., Jajodia, S.: An efficient and unified approach to correlating, hypothesizing, and predicting intrusion alerts. In: Vimercati, S.C., Syverson, P., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 247–266. Springer, Heidelberg (2005)
13. Wang, L., Noel, S., Jajodia, S.: Minimum-cost network hardening using attack graphs. Computer Communications 29(18), 3812–3824 (2006)
14. Wang, L., Yao, C., Singhal, A., Jajodia, S.: Implementing interactive analysis of attack graphs using relational databases. Journal of Computer Security 16(4), 419–437 (2008)

Computationally Perfect Secret Sharing Scheme Based on Error-Correcting Codes

Appala Naidu Tentu[1], Prabal Paul[2], and V.Ch. Venkaiah[3]

[1] CR Rao Advanced Institute of Mathematics, Statistics, and Computer Science
University of Hyderabad Campus, Hyderabad-500046, India
[2] Birla Institute of Technology & Science, Pilani, Goa Campus, GOA-403726, India
[3] Department of Computer and Information Sciences, University of Hyderabad, Hyderabad-500046, India
{naidunit,prabal.paul}@gmail.com, venkaiah@hotmail.com

Abstract. In this paper, we propose a secret sharing scheme for compartmented access structure with lower bounds. Construction of the scheme is based on the Maximum Distance Separable (MDS) codes. The proposed scheme is ideal and computationally perfect. By computationally perfect, we mean, an authorized set can always reconstruct the secret in polynomial time whereas for an unauthorized set this is computationally hard. This is in contrast to some of the existing schemes in the literature, in which an authorized set can recover the secret only with certain probability. Also, in our scheme unlike in some of the existing schemes, the size of the ground field need not be extremely large. This scheme is efficient and requires $O(mn^3)$, where n is the number of participants and m is the number of compartments.

Keywords: Compartmented access structure, computationally perfect, ideal, MDS code, perfect, secret sharing scheme.

1 Introduction

Secret sharing is a cryptographic primitive, which is used to distribute a secret among participants in such a way that an authorized subset of participants can uniquely reconstruct the secret and an unauthorized subset can get no information about the secret. It is a fundamental method used in secure multiparty computations, where various distrusted participants cooperate and conduct computation tasks based on the private data they provide. A secret sharing scheme is called ideal if the maximal length of the shares and the length of the secret are identical. Secret sharing was first proposed by Blakley[2] and Shamir[16]. The scheme by Shamir relies on the standard Lagrange polynomial interpolation, whereas the scheme by Blakley[2] is based on the geometric idea that uses the concept of intersecting hyperplanes.

The family of authorized subsets is known as the access structure. An access structure is said to be monotone if a set is qualified then its superset must also be qualified. Several access structures are proposed in the literature. They include

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 251–262, 2014.

the (t, n)-threshold access structure, the Generalized access structure and the Multipartite access structure. In the (t, n)-threshold access structure there are n shareholders, an authorized group consists of any t or more participants and any group of at most $t-1$ participants is an unauthorized group. Let $\mathbb{U}$ be a set of n participants and let $2^{\mathbb{U}}$ be its power set. Then the 'Generalized access structure' refers to situations where the collection of permissible subsets of $\mathbb{U}$ may be any collection $\Gamma \subseteq 2^{\mathbb{U}}$ having the monotonicity property. In multipartite access structures, the set of players $\mathbb{U}$ is partitioned into m disjoint entities $\mathbb{U}_1, \mathbb{U}_2, \cdots, \mathbb{U}_m$ called levels and all players in each level play exactly the same role inside the access structure.

Compartmented access structure is a multipartite access structure such that all subsets containing at least t_i participants from $\mathbb{U}_i$ for every i, $1 \leq i \leq m$, and a total of at least t_0 participants are qualified to reconstruct the secret. Formally,

$$\Gamma = \{\mathbb{V} \subseteq \mathbb{U} : |\mathbb{V} \cap \mathbb{U}_i)| \geq t_i, \text{for every } i \in \{1, 2, \cdots, m\} \text{ and } |\mathbb{V}| \geq t_0\},$$

where $t_0 \geq \sum_{i=1}^{m} t_i$. This access structure was first proposed by Simmons[17]. It was later generalized it to this form by Brickell[4] and it is now known as the compartmented access structure with lower bounds[23,7]. A secret sharing scheme is a perfect realization of Γ if for all $A \in \Gamma$, the users in A can always reconstruct the secret and for all B not in Γ, the users in B collectively cannot learn anything about the secret, in the information theoretic sense.

1.1 Maximum-Distance-Separable Codes [21]

A linear block code over a field $\mathbb{F}_q$ is a linear subspace of $\mathbb{F}_q^n$, where n is the block length of the code. More generally, a block code C of length n with q^k codewords are called a linear $[n, k]$ code if and only if its q^k code words form a k-dimensional subspace of the vector space consisting of all the n-tuples over the field $\mathbb{F}_q$. An $[n, k, d]$ block code over $\mathbb{F}_q$ is called Maximum Distance Separable (MDS) code if distance $d = n - k + 1$. Two important properties, namely,

- Any k columns of a generator matrix are linearly independent and
- Any k symbols of a codeword may be taken as message symbols, of MDS codes,

have been exploited in the construction of our scheme. It may be noted that for any $k, 1 \leq k \leq q-1$, and $k \leq n \leq q-1$ there is an $[n, k, n-k+1]$ MDS code and an $[q, k, q-k+1]$ extended Reed Solomon code.

1.2 Related Work

Simmons [17] introduced the compartmented access structure and presented ideal secret sharing schemes for some particular examples of this access structure [7]. These constructions are based on the generalization of the geometric method by Blakley [1,6,7]. Ghodasi[8] et.al proposed compartmented schemes for multi-level groups. Constructions of ideal vector space secret sharing schemes for variants of the compartmented access structure and also for some tripartite access structure have been given in [1,5,9,14,23]. Variants of the access structure called compartmented access structure with upper and lower bounds have also been introduced in [4,7,23]. Herranz and Saez [9] offered a family of ideal multipartite access structures that can be seen as a variant of the compartmented access structure. Almost all of these constructions require a huge number of determininants, which can grow exponentially on the number of participants, to be computed [6,4,23]. There are no known schemes for compartmented access structures that circumvent this problem [23].

Tassa [22] and Tassa and Dyn [23] proposed ideal secret sharing schemes, based on Birkhoff interpolation and bivariate polynomial interpolation respectively, for several families of multipartite access structures that contain the multilevel and compartmented ones. One of these schemes, designed for an hierarchical access structure, require the size of the ground field to be bigger than $2^{-k+2}.(k-1)^{(k-1)/2}.(k-1)!$ where k is the minimal size of an authorized subset. Rest of these schemes are perfect in a probabilistic manner. Also there are constraints to be satisfied in assigning identities to the participants. McEliece and Sarwate [13] established that shamir's [16] scheme is closely related to Reed-Solomon codes. Blakley and Kabatianski [3] have showed that ideal perfect threshold secret sharing schemes and MDS codes are equivalent. Pieprzyk and Zhang [10] constructed ideal threshold schemes using MDS codes. Also linear codes have been used earlier in some constructions of threshold schemes [11,15].

1.3 Motivation

The motivation for this study is to come up with a scheme that is efficient, that do not require the ground field to be extremely large, and that offers no restrictions in assigning identities to the users for compartmented access structures with lower bounds [23,7]. The proposed scheme what we call, is computationally perfect. By computationally perfect, we mean, an authorized set can always reconstruct the secret in polynomial time whereas for an unauthorized set this is computationally hard. This is in contrast to the majority of the schemes found in the literature, which are perfect in a probabilistic manner. A scheme is perfect in a probabilistic manner if either an authorized set may not be able to reconstruct the secret or an unauthorized set may be able to reconstruct the secret with some probability [12,23,7].

Remark: A compuational secret sharing scheme [24], in general, tries to reduce the share size by resorting to two distinct fields, one each for secret space and the share space. It may be noted in this context that our concept of computationally perfect secret sharing scheme [18,20,19] is different from that of computational secret sharing scheme. Also, our proposed computationally perfect scheme differs from computational secret sharing schemes in that both the secret and the shares are chosen from the same field.

1.4 Our Results

In this paper, we propose a secret sharing scheme for compartmented access structure with lower bounds. The proposed scheme is ideal and computationally perfect and relies on hardness assumption given in section 2.

Novelty of our scheme is that it overcomes all the limitations present in most of the existing schemes. The size of the ground field, in our scheme, need not be extremely large and there are no restrictions in assigning the identities to the users. It is efficient and require $O(mn^3)$, where n is the number of participants and m number of compartments, computation for Setup, Distribution, and Recovery phases. The construction of this scheme is based on the maximum distance separable (MDS) codes.

Table 1 compares our scheme with some of the secret sharing schemes based on MDS codes.

Table 1.

Scheme	Access Structure	Contribution
McEliece and Sarawate[13]	Threshold access structure	Shamir's[16] scheme is closely related to Reed Solomon code
Blakley and Kabatianski[3]	Threshold access structure	Ideal perfect threshold schemes and MDS codes are equivalent
Pieprzyk and Zhang[10]	Threshold access structure	Constructed ideal threshold schemes using MDS codes
Our proposed scheme	Compartmented access structure with lower bounds	Designed ideal and computationally perfect schemes using MDS codes

Section 2 describes our compartmented secret sharing scheme. Correctness of the scheme is also described in this section. Conclusions are in section 3.

2 Proposed Scheme

Let $U = \bigcup_{i=1}^{m} U_i$ be a set of participants partitioned into m disjoint sets $U_i, 1 \leq i \leq m$. Also, let $|U_i| = n_i$ for all $i \in \{1, 2, \cdots, m\}$. Further, let $k_1, k_2, \cdots, k_m$, and k be positive integers such that $k \geq k_1 + k_2 + \cdots + k_m$ and $k_i \leq n_i$ for all $1 \leq i \leq m$. In addition, let $k'' = \max\ \{k_i : 1 \leq i \leq m\}$, $k' = k'' - \min\ \{k_i : 1 \leq i \leq m\}$, and $n = \max\ \{n_i + 1 : 1 \leq i \leq m\}$.

Assumption. Let $a \in \mathbb{F}_q$ and $f_i : \mathbb{F}_q \rightarrow \mathbb{F}_q$, $1 \leq i \leq 2$, be two distinct one-way functions. Also, let $f_i(a) = b_i$, $1 \leq i \leq 2$. Then the computation of a from the knowledge of either b_1 or b_2 or both is computationally hard.

Overview of the Scheme

Let s be the secret. Choose two one-way functions f_1 and f_2. Also, choose randomly and uniformly $m + 1$ partial secrets $s_i, 1 \leq i \leq m$, and s' such that the secret is the sum of all partial secrets, i.e., $s = \sum_{i=1}^{m} s_i + s'$. Select an $[2N - k, N, N - k + 1]$ MDS code C, where $N = \sum_{i=1}^{m} n_i + 1$. Pick randomly and uniformly $\sum_{i=1}^{m} n_i$ shares, $v_{i,j}, 1 \leq j \leq n_i, 1 \leq i \leq m$, and distribute them to the corresponding participants. Now choose m codewords $C_i, 1 \leq i \leq m$, such that the first component is the partial secret s_i, starting from $(\sum_{j=1}^{i-1} n_j + 2)^{th}$ component n_i elements are the images of the shares of the i^{th} compartment participants under the chosen one-way function f_1, and the rest of the components are arbitrary. $\sum_{j=1}^{m} n_j - k_i + 1$ of these arbitrarily chosen components are made public corresponding to the codeword C_i so that if any k_i of n_i participants of the i^{th} compartment cooperate they can, with the help of the publicized shares, reconstruct the codeword C_i uniquely and hence can recover the first component, s_i, of this codeword, which is the i^{th} partial secret to be recovered.

Now choose yet another, i.e., $(m + 1)^{th}$, codeword C_{m+1} such that the first component is the partial secret s'. Next $\sum_{i=1}^{m} n_i$ components are the images under the second one-way function f_2 of the shares of the participants of the $1^{st}, 2^{nd}$, and so on upto the m^{th} compartment. The last $\sum_{i=1}^{m} n_i - k$ components of the codeword are arbitrary, which are made public so that if any k participants cooperate they can recover the codeword uniquely and hence the partial secret s', which is the first component.

Remark. The components corresponding to the i^{th} compartment participants of the i^{th} codeword C_i are the images of the shares of these participants under the first one-way function f_1. Also the components corresponding to the participants in the $(m + 1)^{th}$ codeword C_{m+1} are the images of the shares of the participants under the second one-way function f_2. Since the two one-way functions are distinct, the knowledge of the components corresponding to the i^{th} compartment of the codeword $C_i, 1 \leq i \leq m$ does not imply the knowledge of the corresponding components of the codeword C_{m+1} and vice-versa.

2.1 Distribution Phase

In this scheme, the dealer considers an $[2N - k, N, N - k + 1]$ MDS code C_2 over the field $\mathbb{F}_q$, where $N = \sum_{i=1}^{m} n_i + 1$. Then the phase consists of the following steps.

1. Choose arbitrarily $s_1, s_2, \cdots, s_m$, and s' from $\mathbb{F}_q$, so that the secret $s = s_1 + s_2 + \cdots + s_m + s'$.
2. Choose m codewords $C_i = (s_i, u_{11}^{(i)}, u_{12}^{(i)}, \cdots, u_{1n_1}^{(i)}, u_{21}^{(i)}, u_{22}^{(i)}, \cdots, u_{2n_2}^{(i)}, \cdots, u_{i-1,1}^{(i)}, u_{i-1,2}^{(i)}, \cdots, u_{i-1,n_{i-1}}^{(i)}, v_{i1}^{1}, v_{i2}^{1}, \cdots, v_{in_i}^{1}, u_{i+1,1}^{(i)}, u_{i+1,2}^{(i)}, \cdots, u_{i+1,n_{i+1}}^{(i)}, \cdots, u_{m1}^{(i)}, u_{m2}^{(i)}, \cdots, u_{mn_m}^{(i)}, u_{m+1,1}^{(i)}, u_{m+1,2}^{(i)}, \cdots, u_{m+1,\sum_{j=1}^{m} n_j - k+1}^{(i)})$, where $v_{ij}^{1} = f_1(v_{ij})$ for $1 \leq j \leq n_i$, $1 \leq i \leq m$ and v_{ij} is a share of the j^{th} participant in the i^{th} compartment, That is v_{ij}^{1} is the image of the share of the j^{th} participant in the

i^{th} compartment under the one-way function f_i. The remaining components are arbitrary.

Remark 1. A way of choosing the $i^{th}, 1 \leq i \leq m$, codeword is as follows:

- Choose an N-vector whose first component is s_i, i^{th} partial secret in the sum of the secret, next n_i components are $v^1_{i1}, v^1_{i2}, \cdots, v^1_{in_i}$ (images of the shares of the i^{th} compartment participants under the one-way function f_1) and rest of the components arbitrary.
- Reduce the generator matrix using elementary row operations so that the n_i columns starting from $(\sum_{j=1}^{i-1} n_j + 2)^{th}$ column to $(\sum_{j=1}^{i} n_j + 1)^{th}$ column form a partial identity matrix.
- Multiply the reduced generator matrix with the N-component vector that was constructed above.

3. Choose another codeword $C = (s', v^2_{11}, v^2_{12}, \cdots, v^2_{1,n_1}, v^2_{21}, v^2_{22}, \cdots, v^2_{2n_2}, \cdots, v^2_{m1}, v^2_{m2}, \cdots, v^2_{mn_m}, u_{m+2,1}, u_{m+2,2}, \cdots, u_{m+2,\sum_{j=1}^{m} n_j - k+1})$, where $v^2_{ij} = f_2(v_{ij})$ for $1 \leq j \leq n_i$, $1 \leq i \leq m$. That is v^2_{ij} is the image of the share of the j^{th} participant in the i^{th} compartment under the one-way function f_2. The remaining components are arbitrary.

   Remark 2. Construction of the codeword C is also similar. i.e.,

   - Construction an N-component vector by taking the first component as s' next n_1 components as $v^2_{11}, v^2_{12}, \cdots, v^2_{1,n_1}$ followed by the n_2 components $v^2_{21}, v^2_{22}, \cdots, v^2_{2n_2}$ and so on upto the n_m components $v^2_{m1}, v^2_{m2}, \cdots, v^2_{mn_m}$ of the m^{th} compartment. By the way v^2_{ij} is the image of the share of the j^{th} participant in the i^{th} compartment under the second one-way function f_2.
   - Reduce the generator matrix using elementary row operations so that the first N columns form the identity matrix.
   - Multiply the reduced generator matrix with the N-component vector that was constructed above.

4. Distribute $v_{i,j}, 1 \leq i \leq m, 1 \leq j \leq n_i$ to the j^{th} participant in the i^{th} compartment.
5. Publicize $(\sum_{l=1}^{m} n_l - k_i + 1)$ of $u^{(i)}_{jk}$'s $1 \leq j \leq m+1, j \neq i, 1 \leq k_i \leq n_i$ as public shares corresponding to C_i, $1 \leq i \leq m$.
6. Similarly, publicize $u_{m+2,j}, 1 \leq j \leq \sum_{j=1}^{m} n_j - k + 1$, as public shares corresponding to the codeword C.

2.2 Recovery Phase

If at least k players such that at least k_t of them from the $t^{th}, 1 \leq t \leq m$, compartment participate then the secret can be recovered. In order to describe the recovery phase, let us assume that $j_t \geq k_t, 1 \leq t \leq m$, players from the t^{th} compartment are participating. Also, let us assume that $l_{t,1}, l_{t,2}, \cdots, l_{t,j_t}, 1 \leq t \leq m$

be the corresponding indices of the participating players. Then the recovery phase consists of the following steps:

1. Fix i such that $1 \leq i \leq m$. Select $(\sum_{j=1}^{m} n_j + 1 - j_i)$ public shares from the $(\sum_{j=1}^{m} n_j + 1 - k_i)$ publicized shares. Let the indices of the selected public shares be $l_{j_i+1,i}, l_{j_i+2,i}, \cdots, l_{(\sum_{j=1}^{m} n_j+1),i}$.
2. Reduce, using elementary row operations, the generator matrix that has the following structure.

   a. If $i = 1$ then $(\sum_{k=1}^{i-1} n_k + 1 + l_{ji})^{th} = (l_{ji} + 1)^{th}$ column has 1 in the j^{th}, $1 \leq j \leq j_i$, row and zeros elsewhere,

   b. $(l_{ji} + 1)^{th}$, column has 1 in the j^{th}, $j_i + 1 \leq j \leq \sum_{k=1}^{m} n_k + 1$, row and zeros elsewhere.

3. Form the message vector
$(v^1_{il_{i,1}}, v^1_{il_{i,2}}, \cdots, v^1_{il_{i,j_i}}, u^{(i)}_{l_{j_i+1,i}}, u^{(i)}_{l_{j_i+2,i}}, \cdots, u^{(i)}_{l_{(\sum_{j=1}^{m} n_j+1),i}})$ by computing
$v^1_{i,l_{i,j}} = f_1(v_{i,l_{ij}})$, $1 \leq j \leq j_i$.
4. Multiply the reduced generator matrix formed in step 2 by the message vector formed in step 3 to arrive at the codeword C_i.
5. First component of the codeword formed in step 4 is s_i, a term in the sum of the secret s.
6. Repeat steps 1 to 5 for every distinct i and recover all the terms except s' in the sum of secret.
7. Select $(\sum_{t=1}^{m} n_t + 1 - \sum_{t=1}^{m} j_t)$ public shares from the $(\sum_{t=1}^{m} n_t + 1 - k)$ publicized shares. Let the indices of the selected public shares be
$l_{m+2,1}, l_{m+2,2}, \cdots, l_{m+2,(\sum_{t=1}^{m} n_t+1-\sum_{t=1}^{m} j_t)}$.
8. Reduce, using elementary row operations, the generator matrix that has the following structure.

   1) $(\sum_{k=1}^{i-1} n_k + l_{ji} + 1)^{th}$ column has 1 in the $(\sum_{k=1}^{i-1} j_k + j)^{th}$, $1 \leq j \leq j_i$, row and zeros elsewhere for every i such that $1 \leq i \leq m$.

   2) $(\sum_{k=1}^{m} n_k + l_{m+2,j} + 1)^{th}, 1 \leq j \leq \sum_{k=1}^{m} n_k + 1 - \sum_{k=1}^{m} j_k$, column has 1 in the $(\sum_{k=1}^{m} j_k + j)^{th}$ row and zeros elsewhere.

9. Form the message vector
$(v^2_{1l_{1,1}}, v^2_{1l_{1,2}}, \cdots, v^2_{1l_{1,j_1}}, v^2_{2l_{2,1}}, v^2_{2l_{2,2}}, \cdots, v^2_{2l_{2,j_2}}, \cdots, v^2_{ml_{m,1}}, v^2_{ml_{m,2}}, \cdots,$
$v^2_{ml_{m,j_m}}, u_{m+2,l_{m+2,1}}, u_{m+2,l_{m+2,2}}, \cdots, u_{m+2,l_{m+2,(\sum_{k=1}^{m} n_k+1-\sum_{k=1}^{m} j_k)}})$.
10. Multiply the reduced generator matrix formed in step 8 by the message vector formed in step 9 to arrive at the codeword C.
11. First component of the codeword formed in step 10 is s'.
12. Recover the secret $s = s' + \sum_{k=1}^{m} s_k$.

Remark. Complexity analysis of the scheme shows that the setup and distribution requires $O((\sum_{i=1}^{m} n_i)^3)$ or $O(\sum_{i=1}^{m} n_i \log q)$ operations. Similarly, the computational requirement of the recovery phase is then $O(m(\sum_{i=1}^{m} n_i)^3)$ or $O(\sum_{i=1}^{m} n_i \log q)$ operations.

2.3 Example

Let $m = 3, n_1 = 2, n_2 = 3, n_3 = 2, k_1 = 1, k_2 = 1, k_3 = 2$ and $k = 4$. So, we consider $[12, 8, 5]$ MDS code over $\mathbb{F}_{19}$. Let the secret s to be shared be 8 and let the two one-way functions f_1 and f_2 be modulo exponentiation of the primitive elements 2 and 3 of $\mathbb{F}_{19}$ respectively.

Distribution Phase

1. Let $s_1 = 5, s_2 = 7, s_3 = 12$ and $s' = 3$. So, that $s = 5 + 7 + 12 + 3 = 8$.
2. Let the chosen 3 codewords be

   $C_1 = (5, 3, 14, 3, 1, 2, 4, 6, 1, 5, 10, 15)$,
   $C_2 = (7, 2, 4, 1, 7, 8, 1, 1, 5, 16, 2, 7)$, and
   $C_3 = (12, 8, 10, 6, 18, 5, 18, 10, 8, 5, 18, 12)$.
   Note: Here the privste shares are chosen as $v_{11} = 13, v_{12} = 7, v_{21} = 18, v_{22} = 6, v_{23} = 3, v_{31} = 9, v_{32} = 17$. The images of the shares under the first one way function will then be $f_1(v_{11}) = 2^{13} = 3, f_1(v_{12}) = 2^7 = 14, f_1(v_{21}) = 2^{18} = 1, f_1(v_{22}) = 2^6 = 7, f_1(v_{23}) = 2^3 = 8, f_1(v_{31}) = 2^9 = 18, f_1(v_{32}) = 2^{17} = 10$. Codeword C_1 is determined by selecting the 8-component vector as $(5, 3, 14, 3, 1, 2, 4, 6)$, reducing generator matrix whose first 8 columns form the identity matrix, and finally multiplying the above 8-component vector with the foregoing reduced generator matrix. Note that the first component of 8-vector is the partial secret s_1, next two components are the images of the private shares under the 1^{st} one-way function f_1 and remaining components are chosen arbitrarily. Codewords C_2 and C_3 are chosen similarly.
3. Let the other chosen codeword be $C = (3, 14, 2, 1, 7, 8, 18, 13, 9, 9, 2, 7)$.

   Note: Choice of the codeword C is similar to that of the above codewords except that the 8-vector consists of the partial secret s' and images of the shares under the 2^{nd} one-way function. Note that $f_2(v_{11}) = 3^{13} = 14, f_2(v_{12}) = 3^7 = 2, f_2(v_{21}) = 3^{18} = 1, f_2(v_{22}) = 3^6 = 7, f_2(v_{23}) = 3^3 = 8, f_2(v_{31}) = 3^9 = 18, f_2(v_{32}) = 3^{17} = 13$.
4. Distribute
   $v_{11} = 13$ to the 1^{st} participant in the 1^{st} compartment,
   $v_{12} = 7$ to the 2^{nd} participant in the 1^{st} compartment,
   $v_{21} = 18$ to the 1^{st} participant in the 2^{nd} compartment,
   $v_{22} = 6$ to the 2^{nd} participant in the 2^{nd} compartment,
   $v_{23} = 3$ to the 3^{rd} participant in the 2^{nd} compartment,
   $v_{31} = 9$ to the 1^{st} participant in the 3^{rd} compartment, and
   $v_{32} = 17$ to the 2^{nd} participant in the 3^{rd} compartment.
5. (a) Publicize 7 of the shares from the list consisting of the 9 components namely 3, 1, 2, 4, 6, 1, 5, 10 and 15 corresponding to the codeword C_1. Let these 7 publicized shares be 3, 1, 2, 4, 1, 5, and 15.
   (b) Similarly, publicize 7 of the shares from the list consisting of the 8 components namely 2, 4, 1, 1, 5, 16, 2, and 7 corresponding to the codeword C_2. Let these 7 publicized shares be 2, 4, 1, 1, 5, 2, and 7.
   (c) Again publicize 6 of the shares from the list consisting of the 9 components namely 8, 10, 6, 18, 5, 8, 5, 18, and 12 corresponding to the codeword C_3. Let these 6 publicized shares be 10, 18, 5, 8, 5, and 12.

6. Finally publicize 4 of the shares from the list consisting of the 4 components namely 9, 9, 2 and 7. Since there are 4 shares to be publicized and there are exactly 4 shares that can be publicized so we publish all these shares.

Recovery Phase. Let $j_1 = 1, j_2 = 2$, and $j_3 = 2$. Also, let $l_{11} = 2, l_{12} = 2, l_{22} = 3, l_{13} = 1, l_{23} = 2$. That is the 2^{nd} participant from the 1^{st} compartment, 2^{nd} and 3^{rd} participants from the 2^{nd} compartment, andboth the participants from the 3^{rd} compartment are participating in recovering the secret.

1. Let $i = 1$. Select 7 public shares from the list of 7 publicized shares. The indices of these selected shares are $l_{21} = 3, l_{31} = 4, l_{41} = 5, l_{51} = 6, l_{61} = 8, l_{71} = 9, l_{81} = 11$.
2. Reduce, using elementary row operations, the generator matrix that has the following structure:

   $(l_{11} + 1)^{th} = 3^{rd}$ column has 1 in the 1^{st} row and zeros elsewhere,
   $(l_{21} + 1)^{th} = 4^{th}$ column has 1 in the 2^{nd} row and zeros elsewhere,
   $(l_{31} + 1)^{th} = 5^{th}$ column has 1 in the 3^{rd} row and zeros elsewhere.
   $(l_{41} + 1)^{th} = 6^{th}$ column has 1 in the 4^{th} row and zeros elsewhere,
   $(l_{51} + 1)^{th} = 7^{th}$ column has 1 in the 5^{th} row and zeros elsewhere,
   $(l_{61} + 1)^{th} = 9^{th}$ column has 1 in the 6^{th} row and zeros elsewhere.
   $(l_{71} + 1)^{th} = 10^{th}$ column has 1 in the 7^{th} row and zeros elsewhere,
   $(l_{81} + 1)^{th} = 12^{th}$ column has 1 in the 8^{th} row and zeros elsewhere.

   After row reduction, we arrive at the following matrix

$$D = \begin{bmatrix} 7 & 6 & 1 & 0 & 0 & 0 & 0 & 15 & 0 & 0 & 3 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 & 0 & 3 & 0 & 0 & 5 & 0 \\ 13 & 2 & 0 & 0 & 1 & 0 & 0 & 6 & 0 & 0 & 4 & 0 \\ 2 & 16 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 17 & 0 \\ 18 & 6 & 0 & 0 & 0 & 0 & 1 & 14 & 0 & 0 & 11 & 0 \\ 5 & 4 & 0 & 0 & 0 & 0 & 0 & 11 & 1 & 0 & 18 & 0 \\ 18 & 8 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 1 & 9 & 0 \\ 13 & 14 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 0 & 10 & 1 \end{bmatrix}$$

3. Form the message vector as $(14, 3, 1, 2, 4, 1, 5, 15)$
4. Multiplying the reduced generator matrix by the message vector formed above, we obtain $(5, 3, 14, 3, 1, 2, 4, 6, 1, 5, 10, 15)$.
5. So, $s_1 = 5$.

   Similarly, let $i = 2$, the message vector is $(7, 8, 2, 4, 1, 1, 2, 7)$ and recovered codeword C_2 is $(7, 2, 4, 1, 7, 8, 1, 1, 5, 16, 2, 7)$, so, $s_2 = 7$ and let $i = 3$, the message vector is $(9, 17, 10, 18, 5, 0, 1, 18)$ and recovered codeword is C_3 is $(12, 8, 10, 6, 18, 5, 9, 17, 0, 1, 11, 18)$, so, $s_3 = 12$.

7. Select 8-5=3 public shares from the 8-4=4 publicized shares. Let the indices of the selected shares be $l_{51} = 1, l_{52} = 2,$ and $l_{53} = 4$.

8. Reduce, using elementary row operations, the generator matrix that has the following structure.

   $(l_{11}+1)^{th} = 3^{rd}$ column has 1 in the 1^{st} row and zeros elsewhere,
   $(n_1+l_{12}+1)^{th} = 5^{th}$ column has 1 in the 2^{nd} row and zeros elsewhere,
   $(n1+l_{22}+1)^{th} = 6^{th}$ column has 1 in the 3^{rd} row and zeros elsewhere.
   $(n_1+n_2+l_{13}+1)^{th} = 7^{th}$ column has 1 in the 4^{th} row and zeros elsewhere,
   $(n_1+n_2+l_{23}+1)^{th} = 8^{th}$ column has 1 in the 5^{th} row and zeros elsewhere,
   $(n_1+n_2+n_3+l_{51}+1)^{th} = 9^{th}$ column has 1 in the 6^{th} row and zeros elsewhere.
   $(n_1+n_2+n_3+l_{52}+1)^{th} = 10^{th}$ column has 1 in the 7^{th} row and zeros elsewhere,
   $(n_1+n_2+n_3+l_{53}+1)^{th} = 12^{th}$ column has 1 in the 8^{th} row and zeros elsewhere.

   After row reduction, we arrive at the following matrix

$$F = \begin{bmatrix} 2 & 15 & 1 & 14 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 0 \\ 11 & 17 & 0 & 17 & 1 & 0 & 0 & 0 & 0 & 0 & 13 & 0 \\ 8 & 9 & 0 & 6 & 0 & 1 & 0 & 0 & 0 & 0 & 9 & 0 \\ 7 & 3 & 0 & 8 & 0 & 0 & 1 & 0 & 0 & 0 & 13 & 0 \\ 13 & 7 & 0 & 13 & 0 & 0 & 0 & 1 & 0 & 0 & 8 & 0 \\ 14 & 3 & 0 & 9 & 0 & 0 & 0 & 0 & 1 & 0 & 6 & 0 \\ 12 & 15 & 0 & 13 & 0 & 0 & 0 & 0 & 0 & 1 & 17 & 0 \\ 10 & 8 & 0 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 14 & 1 \end{bmatrix}$$

9. Form the message vector as $(2, 7, 8, 18, 13, 9, 9, 7)$.
10. Multiplying the reduced generator matrix by the message vector formed above, we obtain $(3, 14, 2, 1, 7, 8, 18, 13, 9, 9, 2, 7)$.
11. So, $s' = 3$.
12. Hence the secret is $s = s_1 + s_2 + s_3 + s' = 5 + 7 + 12 + 3 = 8$.

2.4 Correctness of the Scheme

Following theorems establish that the proposed scheme is ideal and always recovers the secret in polynomial time if and only if the set of participants is an authorized set.

Theorem 1. *The secret can be recovered by the recovery phase described above in polynomial time if and only if the set of participants recovering the secret is an authorized set.*

Proof. Either the codeword C or the codeword C_i for every i, $1 \leq i \leq m$, can be reconstructed by specifying any of its $\sum_{t=1}^{m} n_t + 1$ components. This is because in an $[n, k, d]$ MDS code any k symbols can be treated as message symbols. Further, the reconstruction can be achieved in polynomial time. So, if $j_i \geq k_i$ players cooperate they can recover the codeword C_i and hence s_i in polynomial time with the help of $\sum_{t=1}^{m} n_t + 1 - j_i$ public shares corresponding to the codeword

C_i, for every i, $1 \leq i \leq m$. Similarly, if $j_t \geq k_t$, for every t, $1 \leq t \leq m$, players cooperate they can recover the codeword C and hence s' in polynomial time with the help of $\sum_{t=1}^{m} n_t + 1 - \sum_{t=1}^{m} j_t$ public shares corresponding to the codeword C. Hence, if the set of cooperating participants is an authorized set, they can recover the secret $s = \sum_{t=1}^{m} s_t + s'$ in polynomial time.

Conversely, suppose the set of cooperating participants is an unauthorized set. For the compartmented access structure with lower bounds, an unauthorized set can be of the following types. In one of the types less than k shares are specified in total and in another type k or more shares are specified but there is at least one t, $1 \leq t \leq m$, such that $j_t < k_t$ shareholders only cooperate in the recovery process. It can be visualized that the first type of unauthorized set can not recover s'; whereas the second type of unauthorized set can not recover s_t for which $j_t < k_t$. This is because any $\sum_{t=1}^{m} n_t + 1$ columns are linearly independent. So, the first column of the generator matrix, which corresponds to s_t or s' depending on the codeword C_t, $1 \leq t \leq m$, or C respectively, is not in the span of less than $\sum_{t=1}^{m} n_t + 1$ columns of the generator matrix.

Assume that the unauthorized set is of first type and that $j_t \geq k_t$ for every t, $1 \leq t \leq m$. So, the set can reconstruct all the codewords C_t and hence all the terms s_t for every t, $1 \leq t \leq m$, in the sum of the secret s. But as mentioned previously, it can not reconstruct the codeword C and hence s'. This is because of two distinct one-way functions employed in arriving at the codewords C and C_t, $1 \leq t \leq m$. So, from the hardness assumption, determining the components of C from the corresponding elements of C_t, $1 \leq t \leq m$ is computationally hard.

Above argument can also be extended for the other type of unauthorized set. Therefore, the secret can be recovered in polynomial time if and only if the set of participants recovering the secret is an authorized set.

3 Conclusions

In this paper, we propose a secret sharing scheme for compartmented access structure with lower bounds. This scheme is computationally perfect. This is in contrast to some of the schemes found in the literature, which are perfect in a probabilistic manner. Also, the proposed computationally perfect scheme is ideal. This scheme do not require the ground field to be extremely large and offer no restrictions in assigning identities to the users. Construction of the proposed scheme exploits some of the important properties of MDS codes. This scheme is efficient and require $O(mn^3)$, where n is the number of participants and m is the number of compartments, operations.

References

1. Beimel, A., Tassa, T., Weinreb, E.: Characterizing ideal weighted threshold secret sharing. SIAM J. Disc. Math. 22(1), 360–397 (2008)
2. Blakley, G.R.: Safeguarding cryptographic keys. AFIPS 48, 313–317 (1979)
3. Blakley, G.R., Kabatianski, A.: Ideal perfect threshold schemes and MDS codes. ISIT 1995, 488 (1995)

4. Brickell, E.F.: Some ideal secret sharing schemes. J. Comb. Math. Comb. Comput. 9, 105–113 (1989)
5. Collins, M.J.: A note on ideal tripartite access structures, manuscript available at `http://eprint.iacr.org/2002/193/2002`
6. Farràs, O., Martí-Farré, J., Padró, C.: Ideal multipartite secret sharing schemes. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 448–465. Springer, Heidelberg (2007)
7. Farràs, O., Padró, C., Xing, C., Yang, A.: Natural generalizations of threshold secret sharing. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 610–627. Springer, Heidelberg (2011)
8. Ghodosi, H., Pieprzyk, J., Safavi-Naini, R.: Secret Sharing in Multilevel and Compartmented Groups. In: Boyd, C., Dawson, E. (eds.) ACISP 1998. LNCS, vol. 1438, pp. 367–378. Springer, Heidelberg (1998)
9. Herranz, J., Saez, G.: New results on multipartite access structures. IEEE Proc. Inf. Secur. 153, 153–162 (2006)
10. Pieprzyk, J., Zhang, X.-M.: Ideal threshold schemes from MDS codes. In: Lee, P.J., Lim, C.H. (eds.) ICISC 2002. LNCS, vol. 2587, pp. 253–263. Springer, Heidelberg (2003)
11. Karnin, E.D., Greene, J.W., Hellman, M.E.: On secret sharing systems. IEEE Trans. Inf. Theory, IT-29, 35–41 (1983)
12. Kaskaloglu, K., Ozbudak, F.: On hierarchical threshold access structures. In: IST Panel Symposium, Tallinn, Estonia (November 2010)
13. McEliece, R.J., Sarwate, D.V.: On sharing secrets and Reed Solomon codes. Communications of the ACM 24, 583–584 (1981)
14. Ng, S.-L.: Ideal Secret Sharing Schemes with multipartite access structures. IEEE Proc. Commun. 153, 165–168 (2006)
15. Ozadam, H., Ozbudak, F., Saygi, Z.: Secret sharing schemes and linear codes. In: ISC, Ankara, pp. 101–106 (2007)
16. Shamir, A.: How to share a secret. Comm. ACM 22, 612–613 (1979)
17. Simmons, G.J.: How to (Really) Share a secret. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 390–448. Springer, Heidelberg (1990)
18. Naidu, T.A., Paul, P., Venkaiah, V.C.: Ideal and Perfect Hierarchical Secret Sharing Schemes based on MDS codes. In: Proceeding of International Conference on Applied and Computaional Mathematics, Ankara, Turkey, pp. 256–272 (2012)
19. Tentu, A.N., Paul, P., Vadlamudi, C.V.: Conjunctive Hierarchical Secret Sharing Schemes based on MDS codes. In: Lecroq, T., Mouchard, L. (eds.) IWOCA 2013. LNCS, vol. 8288, pp. 463–467. Springer, Heidelberg (2013)
20. Naidu, T.A., Paul, P., Venkaiah, V.C.: Ideal and Perfect Hierarchical Secret Sharing Schemes based on MDS codes, `eprint.iacr.org/2013/189.pdf`
21. The Theory of Error-Correcting Codes. Macwilliams, Sloane (1981)
22. Tassa, T.: Hierarchical Threshold Secret Sharing. Journal of Cryptology 20, 237–264 (2007)
23. Tassa, T., Dyn, N.: Multipartite Secret Sharing by Bivariate Interpolation. Journal of Cryptology 22, 227–258 (2009)
24. Vinod, V., Narayanan, A., Srinathan, K., Pandu Rangan, C., Kim, K.: On the power of computational secret sharing. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 162–176. Springer, Heidelberg (2003)
25. Yu, Y., Wang, M.: A Probabilistic secret sharing scheme for a compartmented access structure. In: Qing, S., Susilo, W., Wang, G., Liu, D. (eds.) ICICS 2011. LNCS, vol. 7043, pp. 136–142. Springer, Heidelberg (2011)

Warfare Message Communication in Tactical WMN Using HWMP

Jayalakshmi G. Naragund and R.M. Banakar

BVB College of Engineering and Technology, Vidyanagar, Hubli, India

Abstract. In wireless network technologies, Wireless Mesh Network (WMN) is decentralized, low cast and resilient. Hybrid WMN provides minimal configuration and infrastructure to communicate with large community networks like military networks. Since military networks are well planned networks, we can address them as tactical networks. The widely spread fighting forces in military networks have to share information about strategic situations quickly and with better clarity in rapidly changing network. Due to its mobility in terrain topology, network is suffering from suitable routing protocol. Hybrid Wireless Mesh network Protocol (HWMP) is one of the promising routing protocol for such tactical networks. In this paper authors are proposing a framework of analytical model for WMN network, queuing system and delay. The suggested analytical model is also better suited for tactical networks during warefare communication. Using HWMP the analytical delay model is compared with simulation model for 50 nodes. As the simulation proceeds End-to-end delay of delay model is same as simulation model. The authors also focus on analysis of HWMP using military application, against different parameters like Packet Delivery Fraction (PDF), End-to-end delay and routing overhead. The HWMP is compared against well-known protocols of wireless network AODV and DSDV. The results show that PDR and End-to-end delay of HWMP is better than AODV and DSDV as the network size increases. The routing overhead of HWMP is almost nil compared to other two.

Keywords: Tactical networks, Hybrid architecture, Queuing Networks, Ad hoc network.

1 Introduction

Since from two decades, wireless communication has advanced impacting the information technology sector meticulously. Today in the internet era, communicating anywhere and anytime is more dominating. This motivated ISPs (Internet Service Providers) to provide internet through wireless networks with additional cost [1]. Investigations are carried out to obtain network access with reasonable estimates. The result of this is Wireless Mesh Network (WMN) technology, which aspires to access internet with less capital investment.

WMNs have been deployed in many municipal/enterprise area networks. These are used in many applications such as broadband internet access, campus networks, public safety networks, military networks and transportation system etc.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 263–274, 2014.

WMN will also become preferable back bone network with multiple hops and mesh network features.

The WMNs are infrastructure based networks and consists of two types of nodes, which are mesh routers and mesh clients. A wireless mesh router contains additional routing functions to support mesh networking. It is equipped with multiple wireless interfaces and can achieve the same coverage as a conventional router but with much lower transmission power through multi-hop communication [2]. Mesh clients are usual nodes (e.g., desktops, laptops, PDAs, PocketPCs, phones, etc.) equipped with wireless Network Interface Card (NIC) that can connect directly to wireless mesh routers. Users without wireless NICs can also access WMNs by connecting to wireless mesh routers through Ethernet. There are three types of architectures namely infrastructure meshing, client mesh Networking and hybrid mesh networking [2]. Hybrid mesh networking is the combination of infrastructure and client meshing.

WMNs are mainly meant for large community users and scalable networks like military networks. This application uses WMN for sharing information, providing situational awareness and distributes points of intelligence. The fighting force network may be geographically wide spread, asymmetric and rapidly changing. Commanders must regularly be able to access information about situations across the network, with extensive data, voice and video as strategic input. The WMN based tactical networks [3] follow the hybrid mesh architecture. It generally consists of two levels. The top level consists of numerous mesh routers **MRs**, which are forming mesh topology mediating between mesh clients and internet as shown in Fig. 1. To provide low cost deployment, the data-link and physical layer protocols used by **MRs**, are IEEE 802.11, IEEE 802.11s and IEEE 802.16 [4,5]. These have a minimal mobility and they form a backbone of WMN. The bottom level is having mesh clients **MCs**, like lap-tops, mobile phones or PDAs etc., depending on the usage of WMNs by military. As stated in Fig. 1., in tactical networks **MRs** may be deployed on supporting vehicles and a group of size 2 to 4 **MRs** covering a geographical area managed by a Battalion. Since WMNs are envisioned to serve large number of users, efficient routing protocols are necessary. Therefore, the choice of the routing algorithm in a WMN is critical and should be further investigated. The class of hybrid routing protocols are best suited for military network architecture as shown in Fig. 1. These networks use different routing protocols between inter-cluster and intra-cluster nodes. Hybrid Wireless Mesh network Protocol (HWMP) is one of protocol under this class. In HWMP, inter-cluster routing can be activated at regular intervals using proactive protocols, while intra-cluster routing can be on demand using reactive protocols. It aims to provide an optimal solution for dual nature hybrid WMN.

In this paper authors consider the military application for the analysis of HWMP. The contribution of this article is as follows:

- Analytical Model design for network topology, queue in **MRs** and Delay in communication of hybrid WMN.
- Comparison of Delay model.
- Analysis of HWMP using Packet Delivery Ratio (PDF), delay, and overhead.

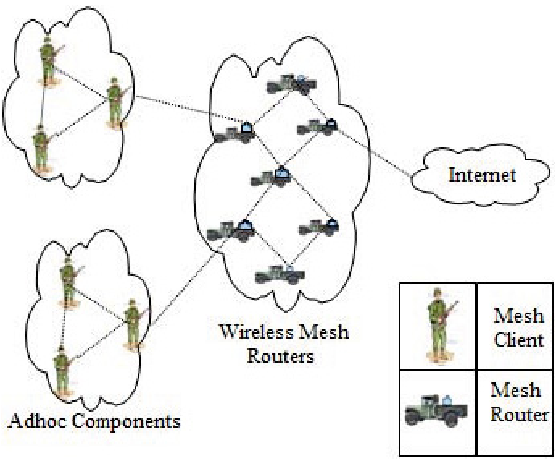


Fig. 1. Architecture of Hybrid WMN in Military application

The rest of the paper is organized as follows: The back ground and related work are discussed in section 2. The analytical model of tactical hybrid WMN is narrated in section 3. Section 4 includes the description of simulation result analysis. Section 5 presents conclusion and future work to be carried out.

2 Literature Survey

In WMN field the researchers are addressing a variety of technical challenges and advanced solutions in the design, implementation and deployment of mesh networks from different aspects such as architectures, protocols, algorithms, services and applications.

Campista and et.al. [6] describe the WMN routing metrics and propose the taxonomy for the main routing protocols. They have done the performance measurements based on the data collected from practical mesh network testbed. Baumann and et.al. [7] design anycast routing protocol called HEAT. This protocol is based on temperature fields and requires communication between neighboring nodes. HEAT has good scalability property due to its fully distributed implementation.

Nabhendra and Alhussein [8] explain the average delay and maximum achievable throughput of random access MAC based WMNs in terms of network parameters. Diffusion approximation method is used to derive expression for end to end delay and throughput. Two tier architecture of WMN is considered for designing the network model and G/G/1 queuing models. Fathemeh and Farid [9] propose the analytical model to evaluate maximum stable throughput of WMNs. They consider the simplified version of IEEE 802.11s MAC protocol for designing queuing network and use stability conditions to present the traffic equations. Tehuang Liu and Wanjiun Liao [10] design analytical model for location dependent throughput and delay. Based on this model, they propose two network strategies to provide fair resource sharing and minimize the end-to-end delay in WMNs. Ping Zhou and et.al. [11] study asymptotic throughput capacity of WMNs. The theoretical results suggest that appropriate number of mesh routers and gateways should be deployed to achieve high throughput.

Malte Cornils and et.al. [12] analyze the performance of HWMP for different parameters like throughput and routing overhead. The protocol uses both proactive and reactive mode of operations. The analysis shows that reactive mode is good for small fraction of root traffic and proactive mode is suitable for increasing fraction of root traffic. Bosung Kim and et.al. [3] propose WMN based tactical network, which uses hybrid architecture and H-AODV routing protocol. Their results show that H-AODV depicts the improved performance of tactical networks than conventional AODV. Nikolaos Peppas and et.al [13] devise hybrid routing algorithm for military applications. They specially consider the moving Unmanned Aerial Vehicles (UAVs) used by Air Force to investigate new grounds. The routing algorithm uses reactive mode between High flying UAVs and proactive mode between low flying UAVs. In their experiment end-to-end delay increases as network grows.

The proactive routing protocols are constantly scanning the network to build and maintain the routes at every node [14]. Reactive routing protocols make use of on demand approach. They establish path between a pair of nodes only when there is need to send data [15].

The hybrid routing protocols combine the functions of proactive and reactive protocols. The idea behind hybrid routing protocol is to use proactive routing mechanism in some area of network at certain times and reactive routing for rest of the network. Some of the hybrid routing protocols are ZRP (Zone Routing protocol), HWMP and WARP (WAve length Routing Protocol). Existing routing protocols of wireless networks are primarily designed for low-end MANETs and are inefficient for WMNs scenario. As we look at the architecture of WMNs, we can conclude that both reactive and proactive routing protocols face problem in providing a solution. A better solution would be to use different routing protocols for the different parts of the network. The HWMP hybrid routing protocol uses reactive mode of routing between mesh clients **MCs** and proactive mode of routing between mesh routers **MRs**.

The authors propose analytical model for tactical hybrid WMN, queue or buffers in routers **MRs** and delay in communication in the next section. The performance of queue mainly depends on arrival rate of packet, busy and idle time of **MRs**. The lemmas proposed in this article are based on these performance parameters. To verify the soundness of suggested analytical model, authors compare the designed delay model with simulation model of hybrid WMN using HWMP.

3 Analytical Model for Tactical WMN and Architecture of HWMP

In this section authors propose an analytical model for networks, queue behavior, End-to-end delay and throughput of tactical hybrid WMN. This section also elaborates architectural design and routing process of HWMP.

3.1 Analytical Model

Proposed analytical model follows the queuing network model of [16]. Queuing network models are used to analyze resource sharing systems like computer networks. They are also powerful tool for evaluating the performance of a system. Authors designed analytical model for hybrid mesh architecture which is more suitable for tactical networks. The analytical model is developed for wireless network, queue delay and End-to-end delay of hybrid WMN.

Network Model. The network model is multiclass Network [16], which includes two classes of packets in the network. The two classes of packets are control and data packets. The size of control packet is negligible compared to the data packet. Data packets are generated by the source node and carry user data. These are not generated by routers. The control packets are used to maintain the network. In analytical model design authors consider only data packets. The network model follows two tier hybrid architecture with **MCs** at lower level and **MRs** at top level. The schematic network model of hybrid WMN is shown in Fig. 2, which consists of uniformly and independently distributed Battalion **MCs**, on unit square area.

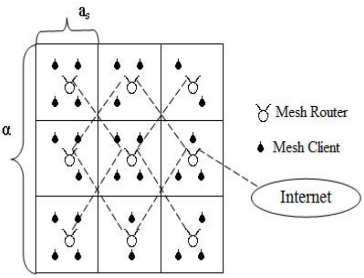


Fig. 2. Network model for tactical Hybrid WMN

Let us consider that α be the length of one side of square area, A be the area of the network and A is calculated using (1).

$$A = \alpha^2 \tag{1}$$

Further A is partitioned into small equal size areas which are called as cells. It includes the Battalion i.e. **MCs** which will come under one **MR**. As shown in Fig 2., the side of a cell is a_s and an area of each cell is depicted by (2).

$$c(i) = a_s^2 \text{ where } 1 \leq i \leq \left(\frac{\alpha}{a_s}\right)^2 \tag{2}$$

Consider $\overline{MR}$ to be the total number of **MRs** given by (3).

$$\overline{MR} = \left(\frac{\alpha}{a_s}\right)^2 \tag{3}$$

MRs are forming mesh network with wireless links. Both **MCs** and **MRs** are single radio nodes and reside in non-overlapping area. Considering real scenario we assumed that **MRs** < **MCs**.

Two different frequency range and bandwidth are used by wireless channel in WMN. The frequency f_c with bandwidth W_c. is used in lower tier to communicate between Battalion **MCs**. Similarly frequency f_r with bandwidth W_r is applied in upper tier to communicate between **MRs**. All wireless nodes in WMN use single radio to transmit and receive the warfare messages. Considering the real time requirement we assume $W_r > W_c$ and $f_r > f_c$.

Queuing Model. The Poisson distribution is convenient mathematical model for many real life queuing systems and it uses the queuing model to describe average arrival rate. Authors consider M/M/1 queuing network with infinite queue length, Poisson distribution for packet arrival rate and each **MRs** having single queue. Queue processing order is Drop tail (FCFS). Since the queue length is infinite WMN is not suffering from drop of packets and blocking of packets. In this section we will discuss the derivation of expressions for different parameters of queuing network model.

Consider R_i mesh router forwards packet from its queue to R_j mesh router queue. We will represent the probability of forwarding packets from R_i and R_j as $P_f(R_i, R_j)$. Consider S_i as a set of neighbors of R_i mesh router and for simplicity we will consider constant number of neighbors to R_i.

Lemma 1. *The arrival rate λ_i at mesh router R_i is given by*

$$\lambda_i = \lambda_{oi} + \sum_{j \in S_i} \lambda_j P_f(R_j, R_i) \tag{4}$$

Proof. The arrival rate λ_i at mesh router R_i is calculated by adding the arrival rate from outside λ_{oi} (i.e. from internet) and arrival rate from all of its neighbors. The arrival rate from all neighbors of R_i is $\sum_{j \in Si} \lambda_j P_f(R_j, R_j)$ and consequently λ_i can be expressed as given in (4).

Corollary 1. *Effective arrival rate λ_{eff} at mesh router R_i is equal to (5).*

$$\lambda_{eff} = \lambda_i \tag{5}$$

Proof. When arrival rate is much more compared to the outgoing rate of packet, blocking or overcrowding of packet will happen at the mesh router. μ_i *is the rate of outgoing packets or the rate of forwarding packets of mesh router* R_i*. i.e.*

$$\begin{array}{l} \lambda_i > \mu_i \;\; \textit{blocking of packets} \\ \lambda_i < \mu_i \; \textit{Unblocking of packets} \end{array}$$

Effective arrival rate is nothing but average number of packet at the mesh router. As mentioned above, queuing model has unlimited buffers and no blocking of packets. So in our model always $\lambda_i < \mu_i$*. According to the property of M/M/1, the effective arrival rate is equal to arrival rate at node (i.e. at* **MRs***) when packets are not blocked. Hence the effective arrival rate* λ_{eff} *at* R_i *is* λ_i*, which leads to (5).*

Lemma 2. *The fraction of time in which* R_i *is busy in forwarding packet and is denoted by* $\rho(R_i)$*.*

$$\rho(R_i) = \frac{\lambda_i}{\mu_i} \tag{6}$$

Proof. One of the main functions of **MR** *is forwarding packets to the neighboring* **MRs***. Thus the* R_i *provides forwarding of packet service to its neighbors. The arrival rate of packets* λ_i *at router* R_i *follows Poisson distribution. The rate at which* R_i *forwards the packets is* μ_i*, which is exponentially distributed. The fraction of time in which* R_i *is busy in forwarding data packets will be the utilization factor of* R_i*. It is the ratio of* λ_i *to* μ_i*, which directs to (6).*

Lemma 3. *The probability of idle time of mesh router* R_i *is* π_i*.*

$$\rho(R_i) = 1 - \frac{\lambda_i}{\mu_i} \tag{7}$$

Proof. Since our model is M/M/1, arrival rate of packet at R_i *is Poisson and service rate are exponentially distributed. Our queuing model can also be modeled as birth death process. Birth rate and death rate at* R_i *is* λ_i *and* μ_i *respectively. As mentioned above* R_i *is unblocking system, so the arrival rate is less than service rate.*

$$\left.\begin{array}{l} \lambda_i < \mu_i \\ \frac{\lambda_i}{\mu_i} < 1 \end{array}\right\} \textit{Unblocking of packets}$$

Using Markov chains concept[16], we can express the probability of idle time of mesh router R_i*,* π_i *as follows*

$$\pi_i = \frac{1}{1 + \dfrac{\lambda_i/\mu_i}{1 - \lambda_i/\mu_i}}$$

By simplifying the above equation, we get (7).

Delay Model. The End-to-end delay is time required to send tactic messages between the military persons of Battalion. We can also formally define the End-to-end delay or latency of WMN as the summation of queuing, propagation and transmission delay. To design the delay model we consider the number of active mesh routers, average queue length and average number of hops traversed by a packet before reaching the destination (Hop count).

The active **MRs** are having packets to send and they will play an important role in the delay calculation, because only their queue delay is considered for the calculation of latency. The events occur at **MRs** are binary in nature, which may or may not occur. In this delay model events considered are arrival of packet and departure of packet from the **MRs**.

Assume hybrid WMN is multihop network and let $\overline{H}$ be the average number of hops traversed by the packet in WMN to reach the destination. In [8] $\overline{H}$ is given by

$$\overline{H} = \frac{1}{P(R_i)} \tag{8}$$

Lemma 4. *The average packet delay at mesh router R_i be $D(R_i)$ and it is expressed as follows:*

$$D(R_i) = \frac{\rho(R_i)\lambda_{eff}}{1-\rho(R_i)} \, 1 \le i \le (\alpha/a_s)^2 \tag{9}$$

Proof. *The average number of packets at R_i will become the average queue length of R_i and we will denote it by K_i. According to [16] K_i is*

$$K_i = \frac{\rho(R_i)}{1-\rho(R_i)} \tag{10}$$

By using Little's law average packet delay at mesh router R_i is

$$D(R_i) = \frac{K_i}{\lambda_{eff}} \; 1 \le i \le (\alpha/a_s)^2$$

Using (10), we will substitute the value of K_i, in the above equation and it leads to (9).

Corollary 2. *The average End-to-end delay $\overline{D}$ is as follows:*

$$\overline{D} = \frac{D(R_i)}{P(R_i)} \quad 1 \le i \le (\alpha/a_s)^2 \tag{11}$$

Proof. *We will use the nature of symmetry and assume average packet delay at all* **MRs** *to be same. As mentioned in [8] the average End-to-end is the product of average number of hops traversed by the packet and queue delay at each* **MRs**.

$$\overline{D} = \overline{H}D(R_i) \quad 1 \le i \le (\alpha/a_s)^2$$

Use (8) and substitute the value of $\overline{H}$ in the above equation, to arrive at (11).

4 Experimental Set Up

The NS-2 simulator is used to analyze the performance of HWMP routing protocol and it is reconfigured after plugged in HWMP protocol patch [17] in network layer. In the experiment set up IEEE 802.11a MAC protocol, the reliable protocol TCP and traffic source FTP is used at datalink, transport and application layers respectively. The HWMP is compared with other two wireless routing protocols AODV and DSDV which are already available in NS-2.

Analysis is carried out in two parts. In the first part designed analytical delay model is compared with simulation model, which is narrated in section 4.1. In the second part HWMP protocol is compared with AODV and DSDV protocols using network scenaries of 50,100 and 200 nodes, which is explained in the section 4.2.

4.1 Analytical Delay Model Comparison

The End-to-end delay calculation in analytical model is done using average hop count $\overline{H}$ and average queue length K_i for a pair of source and destination. In simulation model End-to-end delay is calculated by taking the difference between sent time and received time of packet. The simulation environment of WMN is provided in the Table 1. The average hop count is calculated by retrieving hop count field in the trace file of NS-2 at different time intervals.

Table 1. Simulation environment of WMN

Simulation Area	500m X 500m
Propagation	Radio Model
MAC protocol	IEEE 802.11a
Queue limit	50
Simulation Time	20 sec
nodes in Analytical Model	50
nodes in Simulation Model	50,100,200
protocol in Analytical Model	HWMP
protocol in Simulation Model	AODV, DSDV and HWMP
layer 4 protocol	TCP
Traffic Sources	FTP

Some modifications are done to NS-2 to access the **curq_** variable, which will give the current queue or buffer length of the i^{th} node and it is used in the calculation of average queue length K_i. The analytical model delay is calculated by using equations (10) and (11). The overall arrival rate of network λ assumes three values as shown in Fig. 4(a). for comparing the End-to-end delay, which is obtained from both analytical and simulation model. At the beginning of simulation route to destination will not be set up, so queue length will be more.

As the result of this both simulation and analytical model shows more End-to-end delay at the beginning. After 7 Sec of simulation, the analytical model results subsequently follow the simulation model result.

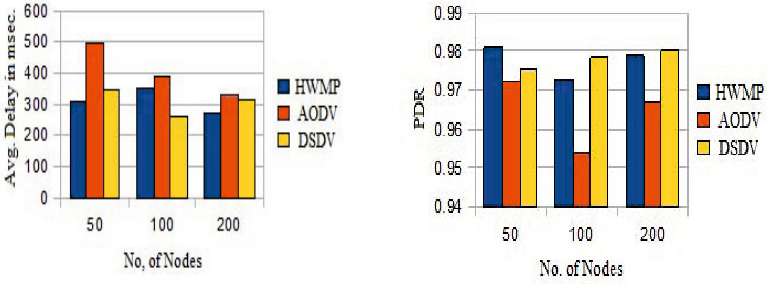
4.2 Simulation Result Analysis

The End-to-end delay $\overline{D}$, routing over head R_H and PDF P_F parameters are obtained using the set of nodes 50, 100 and 200.

Since AODV is reactive protocol, route discovery phase is repeatedly invoked as new nodes enter the adhoc component. So End-to-end delay is more in AODV compared to other to protocols. DSDV is proactive protocol, which has to do routing table updates as adhoc components topography changes and it takes more time for maintaining the routes. Thus DSDV performance is better than AODV as shown in Fig. 3.

As we known HWMP uses both proactive and reactive features and due its mesh topology it utilizes less time in maintaining the routes. Even though networks grow, mesh routers use less time to discover and maintain the routes. Therefore HWMP End-to-end delay is improved compared to other two. As shown in Fig. 3(a). HWMP average delay is 62% less than AODV and 13% less than DSDV for 50 nodes. When network size is 100 nodes HWMP average delay is 11% less than AODV and 27% more than DSDV. Due this result testing is continued considering 200 nodes, where HWMP average delay is 23% less than AODV 18% less than DSDV. The results show that HWMP is having better average delay than AODV and it is competitive for DSDV.

As illustrated in Fig. 3(b). PDF of HWMP is 0.93% more than AODV and 0.6% more than DSDV for 50 nodes. For 100 nodes PDR of HWMP is 1.89% more than AODV and just 0.63% less than DSDV. Similarly PDR of HWMP is 1.24% more than AODV and almost equal to (0.12% less than) DSDV for 200 nodes. We can infer from this analysis that HWMP shows slightly better PDF than AODV and almost equal PDF when compared with DSDV.

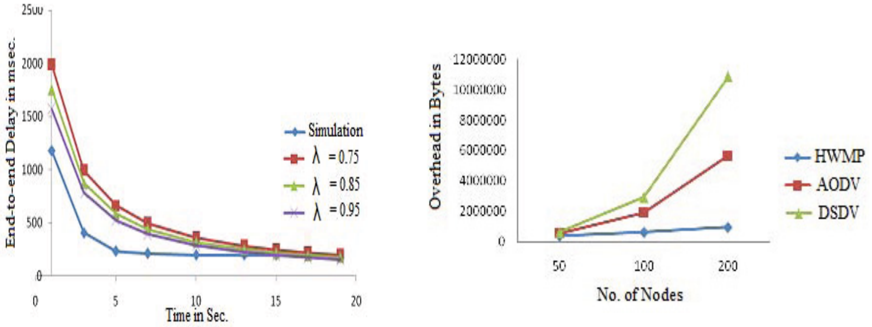


(a) End-to-end delay $\overline{D}$

(b) Packet Delivery Fraction(PDF) P_F

Fig. 3. Analysis of PDF and End-to-end delay

In hybrid mesh network, mesh topology between routers drastically reduces the traffic of control packets used to maintain the routes between source and destination. Even though network size is increased by adding more number of clients, control packets traffic size remains almost same. Fig. 4(b) shows that routing overhead R_H of HWMP is very small compared to other two protocols for 50, 100 and 200 nodes. In 50 nodes network scenario all protocols are having almost same routing overhead. As network grows HWMP shows least overhead compared to other two. HWMP will react to error in the similar way like other wireless protocols. Thus HWMP is best suited protocol for hybrid WMN. Less routing overhead of HWMP allows more user data traffic as network grows.



(a) Analytical and Simulation model comparision

(b) Analysis of routing overhead R_H

Fig. 4. Comparision study and routing overhead R_H

5 Conclusion

The goal of hybrid WMN is to support large mob and thus best suited for tactical military networks. In this paper we designed network, queue, delay and throughput models for hybrid WMN. The End-to-end delay of analytical delay model and simulation model are compared using HWMP. Comparison study reveals that the designed analytical delay model follows simulation model as simulation proceeds. Therefore the designed analytical delay model is appropriate for tactical military networks also. The routing process in HWMP is explained with respect to designed analytical models.

Simulation results shows that routing overhead of HWMP is negligible compared to other two protocols AODV and DSDV. Thus HWMP is efficient routing protocol for hybrid WMN. The End-to-end delay of HWMP is improved as we increased the network size. PDF of HWMP much better than AODV and shows slight improvement over DSDV. We can conclude that when the node density is high, HWMP tends to achieve much better performance, because of less overhead, shorter average routing path and quicker set-up procedure of routing path.

References

1. Nikolaidis, L.: The End of the Internet. IEEE Network (January/February 2008) In: EDITOR'S NOTE
2. Akyildiz, I.F., Wang, X., Wangi, W.: Wireless mesh networks: a survey. Computer Networks and ISDN Systems 47(4), 445–487 (2005)
3. Kim, B., et al.: Tactical network design and simulator with wireless mesh network-based backbone architecture. In: Applications and Technology Conference (LISAT), pp. 1–5 (May 2010)
4. ANSI/IEEE Std 802.11: IEEE Standard for Local and Metropolitan area networks Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. 1999 edition (r2003) edn. (2003)
5. IEEE Std 802.16-2004 (Revision of IEEE Std 802.16-2001): IEEE. IEEE Standard for Local and Metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems (2004)
6. Campista, et al.: Routing metrics and protocols for wireless mesh networks. IEEE Network 22(1), 6–12 (2008)
7. Baumann, Heimlicher, Plattner: Routing in large-scale wireless mesh networks using temperature fields. IEEE Network 22(1), 25–31 (2008)
8. Bisnik, N., Abouzeid, A.: Dealy and throughput in random aceess wireless mesh networks. In: IEEE ICC 2006 Proceedings, pp. 403–408 (2006)
9. Sepehr, F.H., Ashtiani, F.: An analytical model for evaluation of wireless mesh networks. In: International Symposium on Telecommunications, pp. 295–300 (2008)
10. Liu, T., Liao, W.: Location-dependent throughput and delay in wireless mesh networks. IEEE Transactions on Vehicular Technology 57(2), 1188–1198 (2008)
11. Zhou, P., et al.: Asymptotic capacity of infrastructure wireless mesh networks. IEEE Transactions on Mobile Computing 7(8), 1011–1024 (2008)
12. Cornils, M., et al.: Simulative Analysis of the Hybrid Wireless Mesh Protocol (HWMP). In: European Wireless Conference, pp. 536–543 (2010)
13. Peppas, N., et al.: Hybrid routing protocol in wireless mesh networks. In: MILCOM, pp. 1–7 (2007)
14. Abomasan, M., Wysoeki, T., Lipman, J.: Performance investigation on three-classes of MANET routing protocols. In: Asia-Pacific Conference on Communications, October 3-5, pp. 774–778 (2005)
15. Pirzada, A.A., Mcdonald, C., Datta, A.: Performance comparison of trust-based reactive routing protocols. IEEE Transactions on Mobile Computing 5(6), 695–710 (2006)
16. Bolch, G., Greiner, S., de Meer, H., Trivedi, K.S.: Queuing Networks and Markov Chains, ch. 6-7, pp. 212–214, 263–267. John Wiley and Sons (1998)
17. HWMP for NS-2. Wireless Software R&D Group of IITP RAS (February 2009)

Metadata Attribute Based Cipher-Key Generation System in Cloud

R. Anitha and Saswati Mukherjee

Department of Information Science and Technology,
Anna University, Chennai, India

Abstract. With rapid development of cloud computing, more and more IT industries outsources their sensitive data at cloud data storage location. To keep the stored data confidential against untrusted cloud service providers, a natural way is to store only the encrypted data and providing an efficient access control mechanism using a competent cipher key-C_{mxn}, which is becoming a promising cryptographic solution. In this proposed model the cipher key is generated based on attributes of metadata. The key problems of this approach includes, the generation of cipher key and establishing an access control mechanism for the encrypted data using cipher key where keys cannot be revoked without the involvement of data owner and the metadata data server (MDS), hence makes data owner feels comfortable about the data stored. From this study, we propose a novel Metadata Attribute Based Key Generation scheme for cloud data security system by exploiting the characteristic of the data stored. We have implemented a security model that incorporates our ideas and evaluated the performance and scalability of the secured model.

Keywords: Cloud Security, Cipher-key, Data Storage, Metadata.

1 Introduction

Cloud computing has become the most attractive field in industry and in research. The requirement for cloud computing has increased in recent days due to the utilization of the software and the hardware with less investment [5]. A recent survey regarding the use of cloud services made by IDC, highlights that the security is the greatest challenge for the adoption of cloud computing technology [6]. The four key components of data security in cloud computing are data availability, data integrity, data confidentiality, and data traceability. Data traceability means that the data transactions and data communication are genuine and that the parties involved are said to be the authorized persons [7]. Several studies show that data traceability mechanism have been introduced, ranging from data encryption to intrusion detection or role-based access control, doing a great work in protecting sensitive information. However, the majority of these concepts are centrally controlled by administrators, who are one of the major threats to security [8]. Further in the existing system, all authentications are done using cloud user's identity and must be validated by the central authority on the behalf of the cloud service providers. Hence Encryption paves the way for securing the data stored at cloud environment. The existing encryption proposals do not adequately address several

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 275–287, 2014.

important practical concerns that emerge in encryption process. The practitioner often need the flexibility to encrypt the whole data only by using portions of a message in order to create the key for encryption, yet still encrypts the entire message. Such keys can be created using the data associated with the file known as associated data. Finally, some schemes require all parties to agree on the same parameters, such as the common parameters which makes any changes to the security parameter encryption scheme behaves quite difficult. Such associated data can be metadata. In some modern distributed file systems, data is stored on devices that can be accessed through the metadata, which is managed separately by one or more specialized metadata servers [1]. Metadata is a data about data and it is structured information that describes, explains, locates, and makes easier to retrieve, use, or manage an information resource. The metadata file holds the information about a file stored in the data servers. In cloud computing, the users will give up their data to the cloud service provider for storage. The data owners in cloud computing environment want to make sure that their data are kept confidential from outsiders, including the cloud service provider. To this end, encryption is perhaps the most successful mechanism used. The most prominent issue in a centralized administration using encrypted data is key generation and key handling since all such keys are available to the centralized authority and hence can be easily breached. When the secret key is generated in a single space, the system can be easily attacked, either by an external entity or even by the internal entity. Most of the existing cloud encryption schemes are constructed on the architecture where a single trusted (TPA) third party authority has the power to secure the secret data stored at the cloud servers. The major drawbacks of the prevailing systems are that the data stored is not fully secured because the entire security is taken care by a single space. Hence in order to overcome these key related issues, this research proposes a novel method of cipher key generation and key handling mechanism using the metadata attributes. The proposed method takes away the necessity of having a centralized control over the encryption and decryption technique. The specification of deciding the key is based on the metadata attribute in the metadata server as well as the user key. In the proposed system, the key generation and issuing protocol is developed using user key and metadata attributes. The model also makes data owner confident about the complete security of the data stored, since the encryption and decryption keys cannot be compromised without the involvement of data owner and the MDS.

The contributions in the paper can be summarized as follows:

1. This paper proposes a model to create a cipher key C_{mxn} based on the attribute of metadata stored using a modified feistel network and supports users to access the data in a secured manner.

2. It also proposes a novel security policy which involves the data owner and the MDS by means of key creation and sharing policies. Hence the model prevents unauthorized access of data.

The rest of the paper is organized as follows: Section 2 summarizes the related work and the problem statement. Section 3 describes the system architecture model and discusses the detailed design of the system model. Section4 describes the modified feistel network structure design and issues of the proposed model. The construction of the CTC and generation of cipher key is explained in section 5. The performance evaluation based on the prototype implementation is given in Section 6 and Section 7 concludes the paper.

2 Related Works

The Related work discusses about the previous work carried out in the area of cloud security and we have also discussed about how metadata is used in cloud computing environment.

2.1 Metadata in Distributed Storage Systems

Recently much of the work is being pursued in data analytics in cloud storage [1] [2]. Abhishek Verma et al. [3] have proposed metadata using Ring file system. In this scheme metadata for a file is stored based on hashing its parent location. Replica is stored in its successor metadata server. Yu Hua et al. [4] have proposed a scalable and adaptive metadata management in ultra large scale file systems. Michael Cammert et al. [1] distinguished metadata into static and dynamic metadata. He has suggested publish-subscribe architecture, enables a SSPS to provide metadata on demand and cope up with metadata dependencies. R.Anitha et al. [5] has described that the data retrieval using metadata in cloud environment is less time consuming when compared to retrieving a data directly from the data server.

2.2 Security Schemes

Chirag Modi et al. [10] discussed a survey paper where they discussed about the factors affecting cloud computing storage adoption, vulnerabilities and attacks, and identify relevant solution directives to strengthen security and privacy in the Cloud environment. They discuss about the various threats like abusive use of cloud computing, insecure interfaces, data loss and leakage, identity theft and metadata spoofing attack. J. Ravi Kumar et al. [9] shows that third party auditor is used periodically to verify the data integrity stored at cloud service provider without retrieving original data. In this model, the user sends a request to the cloud service provider and receives the original data. If data is in encrypted form then it can be decrypted using his secret key. However, the data stored in cloud is vulnerable to malicious attacks and it would bring irretrievable losses to the users, since their data is stored at an untrusted storage servers. Aguilera et al. [11] has explained about the block level security. R.Anitha al. [12] has proposed a new method of creating a cipher key using modified feistel function. As the metadata attributes changes then there exists a significant change in the cipher key. As the metadata changes every time the key generation process becomes active and a new cipher key is generated. As the metadata attributes changes then there exists a significant change in the cipher key. As the metadata changes every time the key generation process becomes active and a new cipher key is generated. She has also discussed about the avalanche effect of cipher key created using modified feistel function. R.Anitha al. [13] has also proposed a new method providing steganography technique for providing security to the data stored at the cloud environment.

3 System Architecture

The architecture diagram of the proposed system model is shown in Fig.1. The system model proposes security to the data using Cipher Tree Chaining (CTC) network where the metadata attributes are taken as input in the form of matrices. In this model the user uploads the encrypted file using the key K_1. The metadata for the file is created. The metadata creation is based on Dublin Core Metadata Initiative standard. When the user uploads a data file, the file is analyzed and based on DCMI design relevant attributes, such as filename, date of uploaded, size of the file, type of the file, owner and keyword, are extracted and stored as a metadata file. Based on the metadata created, the cipher key C_{mxn} ($CT_{N(mxn)}$) which is the final value from the CTC Network is generated after progressing through several matrix obfuscations. The Metadata server sends the cipher key C_{mxn} to the user. Using C_{mxn} as key, the user encrypts the key K_1 and generates K_2. While downloading the file, the key K_2 and C_{mxn} is used to retrieve K_1 and the file is thus decrypted. This model proposes a novel method of generating cipher key using CTC network which uses the matrix based hashing algorithms. This model provides high strength to the cipher, as the encryption key induces a significant amount of matrix obfuscation into the cipher. The avalanche effect discussed shows the strength of the cipher C_{mxn}.

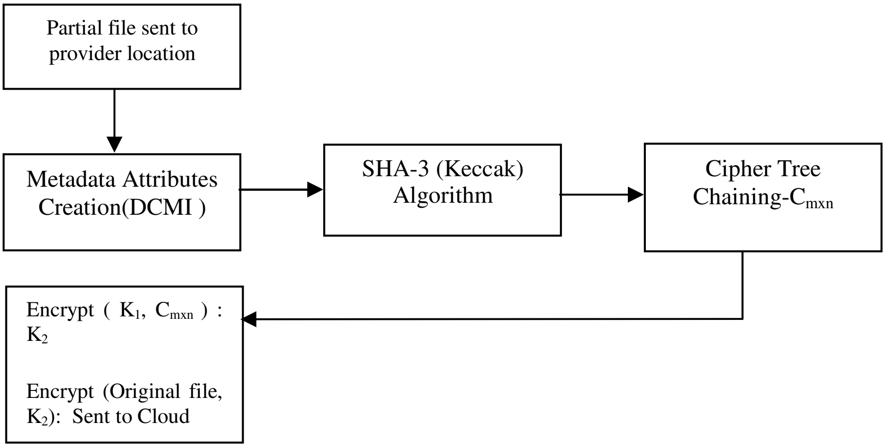
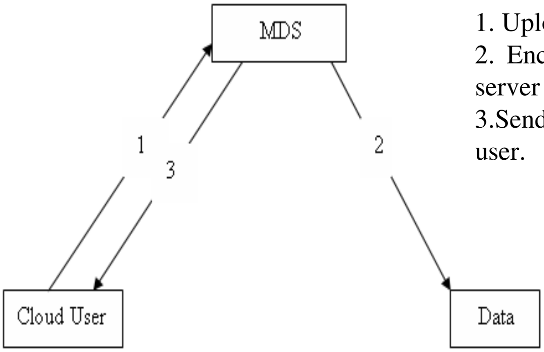

Partial file sent to provider location
Metadata Attributes Creation(DCMI)
SHA-3 (Keccak) Algorithm
Cipher Tree Chaining-C_{mxn}
Encrypt (K_1, C_{mxn}) : K_2
Encrypt (Original file, K_2): Sent to Cloud

Fig. 1. Architecture Diagram

Fig. 2 below provides an overview of the general flow of data file and the cipher key Cmxn. When an user upload a file, partial file is sent to the provider location thereby generating the cipher key. Using the cipher key Cmxn and the key K_1, the user generates a key K_2. Using K_2 user encrypts the whole data file and sends the data to the cloud storage. Hence the file stored is in encrypted form at the cloud storage. Hence without the involvement of user and the MDS the original file cannot be reverted.



1. Uploads the Encrypted file
2. Encrypted File sent to the Data server
3.Sends the Cipher key C_{mxn} to the user.

Fig. 2. Overall Flow process of file Upload and Download using Security key

The step by step method of system functionalities are given below:

- Partial file send to the provider
- Construction of Cipher Tree Chaining Network
- Generation of Cipher key Cmxn and sends back to the user
- User uses Cmxn and some key K_1 to generate K_2.
- User encrypts file using K_2 and sends the encrypted file.
- User destroys K_2 and saves K_1 safely
- To access the file, user asks for Cmxn
- The provider sends Cmxn and the encrypted file
- User uses Cmxn and the key K_1 to generate K_2 again.
- User decrypts file using K_2.

4 Cipher Tree Chaining Network

Cipher tree chaining is a special class of iterated block ciphers where the cipher key is generated from the attributes of metadata by repeated application of the same transformation or round function. Development of the cipher key “C_{mxn}” using Cipher Tree Chaining is described below. This paper proposes a procedure for generating the cipher key “C_{mxn}” based on matrix manipulations. The proposed cipher key generation model offers two advantages. First, the use of the generated key C_{mxn} is simple enough for any user to combine with K_1 and produce K_2. Secondly, due to the complexity of the cipher key, the model produces a strong avalanche effect making many values in the output block of a cipher to undergo changes if even one value changes in the secret key. In the proposed model cloud security model, matrix based cipher key encryption mechanism has been introduced and key avalanche effects have been observed. Thus the cloud security model is improved by introducing a novel mechanism using cipher key chaining network where the cipher key C_{mxn} is generated.

Procedure for generating Cipher Key C_{mxn}:
The cipher key generation procedure is based on a matrix, initialized using secret key and HMAC SHA-3 algorithm. The values used in one round of chain are taken from the previous round. The selection of rows and columns for the creation of matrix is based on the number of attributes of the metadata and the secret message key matrix "M_{K1}" and the other functional logic as explained in the following subsections.

4.1 Data Preprocessing

Data preprocessing is a model for converting the metadata attributes into matrix form using the SHA-3 cryptographic algorithm, containing m-rows and n-columns, where m is the number of attributes of the metadata and n takes the size of the SHA-3 output. The matrix M_{mxn} is splitted into binary tree matrix as explained in the Figure 3 until the column value of the matrix takes an odd number. For convenience the leaf nodes are termed as ED_1, ED_2 … $ED_{N.}$ which are in the matrix form. The number of leaf node depends on the number of attributes of metadata. The matrix obfuscation is carried out in order to make the hacker opaque. The leaf node matrices are fed as an input to the cipher tree chain. Figure 4 represents the flow chart for generating the cipher key C_{mxn}.

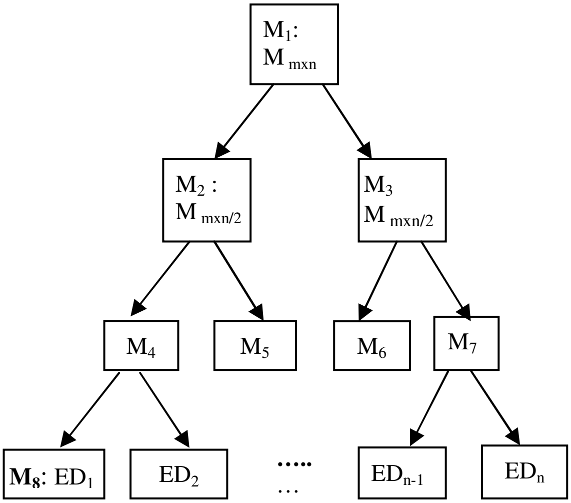


Fig. 3. Model for Data Fragmentation

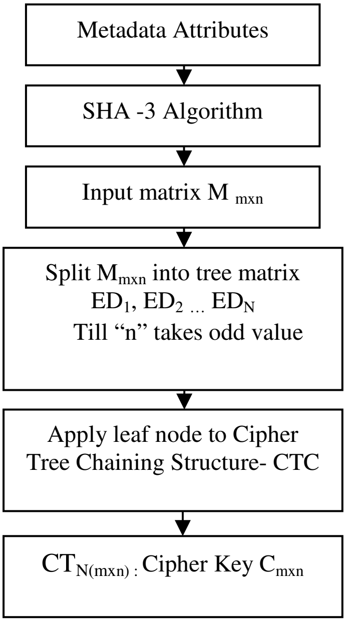

Metadata Attributes
SHA -3 Algorithm
Input matrix M mxn
Split Mmxn into tree matrix ED1, ED2 ... EDN Till "n" takes odd value
Apply leaf node to Cipher Tree Chaining Structure- CTC
CTN(mxn) : Cipher Key Cmxn

Fig. 4. Flow Chart for Generation of Cipher Key: C_{mxn}

4.2 Cipher Tree Chaining Structure

The Matrix M_{mxn} which is SHA-3 value of the metadata attributes is considered as the initial node value of the cipher tree structure. SHA-3 is used in this model because of its irreversible in nature. Specifically, it is not vulnerable to length extension attacks. The M_{mxn} which is considered as an $M_{1mxn}$is splitted into binary tree structure $M_{2mxn/2}$, $M_{3mxn/2}$. Further the matrix is splitted into $M_{3mxn/4}$ and $M_{4mx/4.}$ The process is repeated till the value of "n" becomes odd. After splitting the leaf node is considered as ED_1 which is the initial vale given to the first block of the cipher. Addition operation is performed using $S_{k(mxn)}$ and ED_1, intermediate value is generated. Further the intermediate value $D_{i(mxn)}$ given as input to the HMAC-SHA3 by using key K_1, which is a key from the user. Hence the $CT_{1(mxn)}$ is generated. The development of the cipher key in the cipher tree chaining network is carried out through number of rounds until the all the leaf node values are supplied as input to the CTC structure. In this symmetric block ciphers, matrix obfuscation combined with HMAC operations are performed in multiple rounds using the key matrix. The function secured key $S_{K(mxn)}$ plays a very important role in deciding the security of block ciphers. The generation process of $S_{K(mxn)}$ is explained in Figure6. Fig.5 below represents the one round cipher tree chaining network structure.

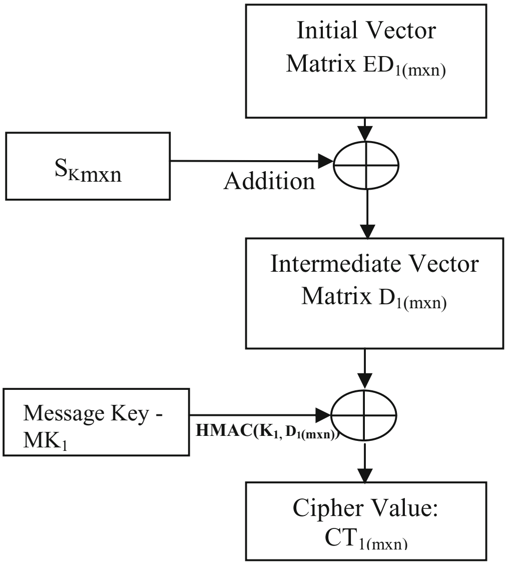

Initial Vector Matrix $ED_{1(mxn)}$
S_{Kmxn}
Addition
Intermediate Vector Matrix $D_{1(mxn)}$
Message Key - MK_1
HMAC(K_1, $D_{1(mxn)}$)
Cipher Value: $CT_{1(mxn)}$

Fig. 5. One Round of Cipher Tree Chaining Network

Definition of Cipher Block

A block cipher is a parameterized deterministic function mapping n-bit plaintext blocks to n-bit cipher text blocks. The value n is called the block length. In the proposed block cipher input is the first leaf node value K be a hidden random seed, then the cipher function is defined as $C(MD_X,S_K,K) = CTx$ where C is a cipher tree function. The procedure for developing the function is described below. The function f is considered to be varied based on the leaf node value and the HMAC- SHA3.In this CTC network structure, each round depends on the previous round value i.e.

$$\text{Round (Ri)} = (\text{Ri-1}, \text{C}(R_{i-2}))$$

The above formula shows that a small change in one round affects the entire CTC network. The number of rounds depends on the number of leaf nodes.

4.3 Generation of Secured Key $S_{K(mxn)}$

The second level of security in the CTC is provided using the secured key $S_{K(mxn)}$. The generation of the key $S_{K(mxn)}$ is as shown in Fig.6. A Secured Key is created based on the metadata attributes. The attribute value of metadata and the user key K_1 are used to generate the secured key. The attribute values are hash coded using the key from the user and generates X_1 and the process is repeated for all the metadata attributes. The cumulative values of the HMAC-SHA3 outputs are considered as the secured key $S_{K(mxn)}$. i.e. X_1 takes the first row of the matrix, X_2 takes the second row of the matrix and so on. Hence the secured key cannot be compromised without the involvement of the user and the metadata attributes which further strengthens the cipher key.

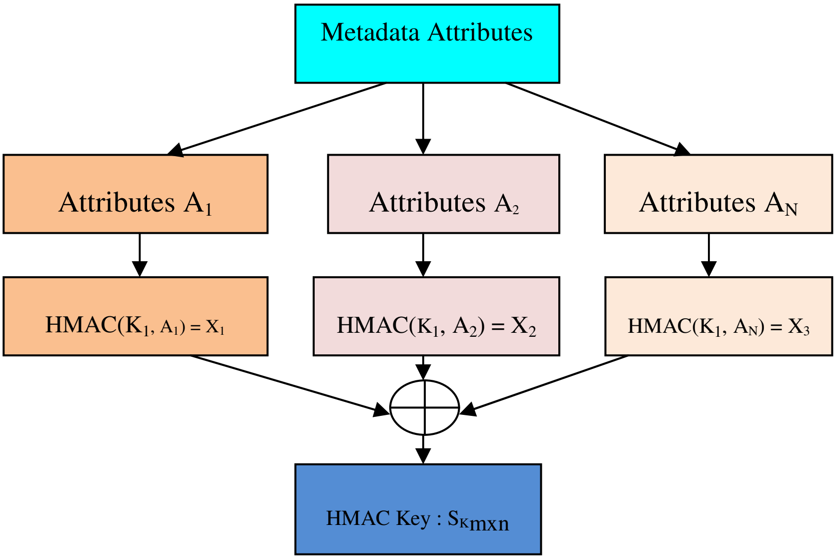


Fig. 6. Generation of Secured Key $S_{K(mxn)}$

5 Construction of Cipher Tree Chaining Network and Generation of Cipher key Cmxn

The construction of the cipher tree chaining network and the cipher key generation is as shown in Figure 7. The leaf node value ED_1 which is the initial value is provided as input to the initial block of the cipher block chaining structure. Addition operation of $S_{k(mxn)}$ and ED_1 is carried out and intermediate value is generated. Further the intermediate value $D_{i\ (mxn)}$ along with the Message key M_{k1} which is decided based on the length of the attributes of metadata. The attribute whose is length is more compared to other attributes of the same file is considered as M_{k1} and supplied to the HMAC-SHA3 which produces $CT_{1(mxn)}$ which is further provided as a secret key to the next block. Thus throughout the cipher block chaining network current block depends on the previous block output, thus maintaining a chain process. The process is continued till the leaf node becomes null. In this symmetric block ciphers, the entire process is carried in matrix form which holds a major strength to the key generation. The strength of the cipher key determines the strength of the proposed security model.

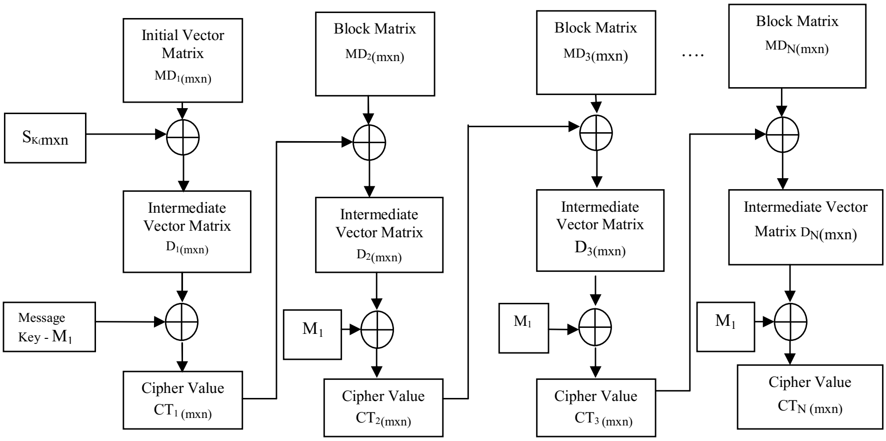


Fig. 7. Cipher key generation using Cipher Tree Chaining Structure

5.1 Pseudo Code for Creation of Cipher Key Cmxn

Algorithm 1: Pseudo code Creation of Cipher Key C_{mxn}

```
Begin

 1.Read Metadata attribute
 2.Apply SHA-3
 3.  Generate Matrix $M_{mxn}$, split the matrix, and generate cipher tree
 4.  Root value of cipher tree splitted into child nodes
      For i = 1 to n Repeat till n / 2 = 1
     Begin
       4.1 Leaf node values =  $MD_{1mxn}$
       4.2 Add $MD_{1mxn}$, $S_{k\ (mxn)}$  $[MD_{1mxn} + S_{k(mxn)}] = D_{1mxn}$
       4.3 Apply HMAC ($D_{1mxn}$ , $M_{K1}$) = $CT_{1mxn}$
       4.4  Apply $CT_{1mxn}$ as input to the next block as secret key.
 5.  Repeat the step till leaf node becomes null
 6. Write(C) Cipher key C = $CT_{N(mxn)}$
End
```

5.2 Analysis of Cipher Key C_{mxn}: Avalanche Effect

The Cipher Tree Chaining network holds good for the avalanche effect as each round depends on the previous round value. Avalanche effect is an important characteristic for encryption algorithm. This characteristic is seen that a small change in the metadata attribute will have the effect on its cipher key which shows the efficacy of the cipher key, i.e. when changing one bit in plaintext and will change the outcome of at least half of the bits in the cipher text. The discussion of avalanche effect establishes that changing only one bit in the input leads to a large change in the cipher key. Hence it is hard for an attacker to perform any analysis of the cipher key.

$$\text{Avalanche Effect} = \frac{\text{Number of values changed in the Cipher Key } C_{mxn}}{\text{Total Number of values in the Cipher Key } C_{mxn}}$$

The Avalanche effect discusses about the variations in the cipher key C_{mxn} small change in the metadata attribute will have a direct impact on the cipher key. The strength of the key resides in the dynamism of the key, i.e. even for a small change in the metadata attribute which is calculated by avalanche effect. In the case of high-quality block ciphers, such a small change in either the key or the plain text should cause a drastic change in the cipher key. If a block cipher or cryptographic hash function does not exhibit the avalanche effect to a significant degree, then it has poor randomization, and thus input can be predicted, being given only the output. This may be sufficient to partially or completely break the algorithm. Thus, the avalanche effect is a desirable condition from the point of view of the designer of the cryptographic algorithm or device. Thus the proposed model has been verified with the avalanche effect seriously.

6 Implementation and Results Discussion

The experiments have been carried out in a cloud setup using eucalyptus which contains cloud controller and walrus as storage controller. These tests were done on 5 node cluster. Each node has two 3.06 GHz Intel (R) Core TM Processors, i-7 2600,CPU @ 3.40GHZ, 4 GB of memory and four 512 GB hard disks, running Eucalyptus. The tests used a 500 files of real data set, uploaded into the storage and then downloaded based on the user's requirement. The experimental results show that the model provides a complex cipher key C_{mxn} which adequately strengthens the data stored. Results demonstrate that our design is highly complete in nature and the time taken for generating the cipher key is less compared to the existing algorithms. Performance Analysis metrics is done based on the experimental set up as described above. To the best of the domain knowledge obtained due to a wide literature survey on cloud-based performance analysis methodologies and tools, the performance analysis metrics useful for analyzing the cloud security are listed and the comparison results are given.

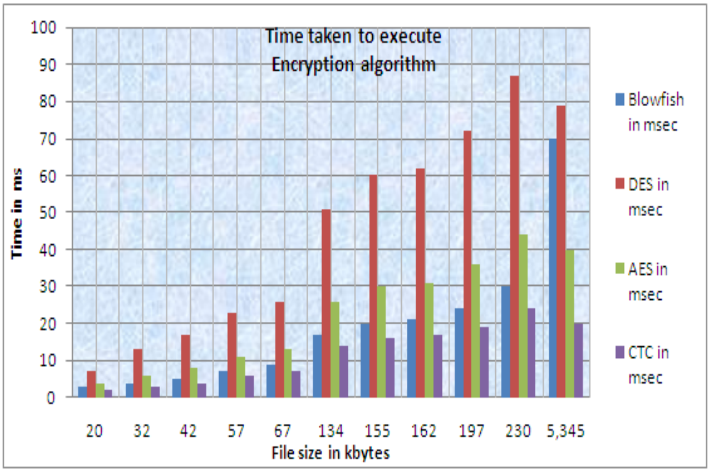


Fig. 8. Comparison of encryption algorithms execution time

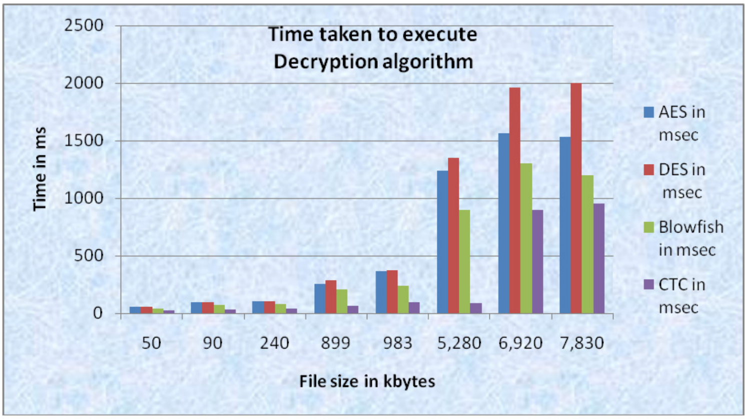


Fig. 9. Comparison of Decryption algorithms execution time

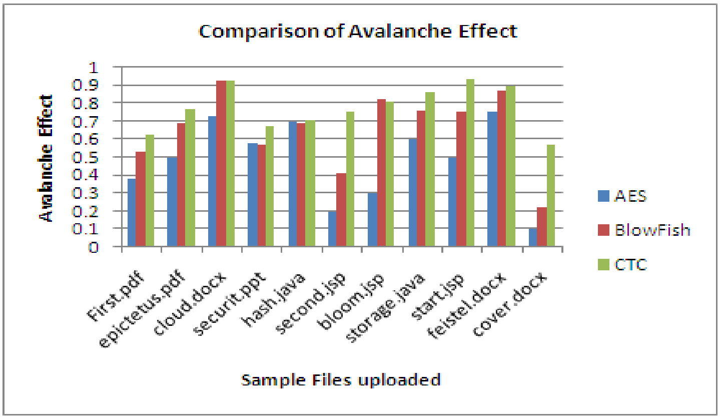


Fig. 10. Comparison of avalanche effect of exixting algorithms

Table 1. Comparison Table of various existing algorithms

Features Analysed	Algorithms				
	DES	AES	Two Fish	Blow Fish	CTC Network
Created By	IBM in 1975	Joan Daemen & Vincent Rijmen in 1998	Bruce Schneier in 1993	Bruce Schneier in 1993	2013
Algorithm Structure	Feistel Network	Substitution-Permutation Network	Feistel Network	Feistel Network	CTC
Rounds	16	10, 12 or 14	16	16	4
Key Size	56 bits	128 bits, 192 bits, 256 bits	128 bits, 192 bits or 256 bits	32-448 bit in steps of 8 bits. 128 bits by default	256*6 bits
Type	Block cipher	Block cipher	Block cipher	Block cipher	Block cipher
Block Size	64 bits	128 bits	128 bits	64 bits	64 bits
Algorithm Running Time kbytes/msec	5kb/msec	2kb/ msec	150kb/ msec	190kb/msec	250kb/msec
Key Strength	Low	Low	High	Very High	Very High (due to Rate of increase in avalanche effect)
Existing Cracks	Brute force attack, differential crypanalysis, linear cryptanalysis, Davies' attack	Side channel attacks	Truncated differential cryptanalysis	Second-order differential attack	NIL –(work in progress)
Avalanche Effect	Less	Less	Moderate	Moderate	High

7 Conclusion

This paper investigates the problem of data security in cloud data storage where the data is stored away from the user. The problem of privacy of data stored has been studied and an efficient and secured protocol is proposed to store data at the cloud storage servers. We believed that the data storage security in cloud era is full of challenges especially when the data is at rest and at the data location. Our method provides privacy to the data stored and the challenge in constructing the security policy, involves both the data owner as well as the MDS to store and retrieve the original data. As the key is in matrix form, it provides major strength to the proposed model. The model also makes data owner confident of the security of the data stored in the centralized cloud environment, since the encryption and decryption keys cannot be compromised without the involvement of both the data owner and the MDS.

References

[1] Cammert, G.M., Kramer, J., Seeger, B.: Dynamic Metadata Management for Scalable Stream Processing Systems. In: IEEE International Conference on Data Engineering Workshop, pp. 644–653 (2007)

[2] Wu, J.-J., Liu, P., Chung, Y.-C.: Metadata Partitioning for Large-scale Distributed Storage Systems. In: IEEE International Conference on Cloud Computing (2010)

[3] Verma, A., Venkataraman, S., Caesar, M., Campbell, R.: Efficient Metadata Management for Cloud Computing Applications. In: International Conference on Communication Software and Networks (2010)

[4] Hua, Y., Yifeng, Jiang, H., Feng, D., Tian, L.: Supporting Scalable and Metadata Management in Ultra Large Scale File Systems. IEEE Transactions on Parellel and Distributed Systems 22(4) (2011)

[5] Anitha, R., Mukherjee, S.: A Dynamic Semantic Metadata Model in Cloud Computing. In: Krishna, P.V., Babu, M.R., Ariwa, E. (eds.) ObCom 2011, Part II. CCIS, vol. 270, pp. 13–21. Springer, Heidelberg (2012)

[6] Kuyoro, S.O.: Ibikunle.F and Awodele. O.: Cloud Computing Security Issues and Challenges. International Journal of Computer Networks 3(5), 247–255 (2011)

[7] Mathew, A.: Survey Paper on Security & Privacy Issues in Cloud Storage Systems. In: Electrical Engineering Seminar and Special Problems 571B (2012)

[8] Heurix, J., Karlinger, M., Neubauer, T.: Perimeter – Pseudonymization and Personal Metadata Encryption for Privacy-Preserving Searchable Documents. In: International Conference on Health Systems, vol. 1(1), pp. 46–57 (2012)

[9] Ravi kumar, J., Revati, M.: Efficient Data Storage and Security in Cloud. Proc. International Journal of Emerging trends in Engineering and Development 6(2) (2012)

[10] Modi, C., Patel, D., Borisaniya, B., Patel, A., Rajarajan, M.: A survey on security issues and solutions at different layers of Cloud computing. Journal of SuperComputers, 561–592 (2013)

[11] Aguilera, M.K., Lillibridge, M., Maccormick: Block-Level Security for Network-attached disks. In: The 2nd Usenix Conference on File and Storage Technologies, pp. 159–174 (2003)

[12] Anitha, R., Paramjothi, P., Yogesh, P., Mukherjee, S.: Data Storage Security in Cloud using Metadata. In: International Conference on Machine learning and Computer Science, pp. 35–39 (2013)

[13] Anitha, R., Paramjothi, P., Mukherjee, S.: Security as a Service using Data Steganography in Cloud computing. In: International Conference on Cloud Security Management, pp.81–89 (2013)

Cryptanalysis of Yang et al.'s Digital Rights Management Authentication Scheme Based on Smart Card

Dheerendra Mishra and Sourav Mukhopadhyay

Department of Mathematics
Indian Institute of Technology Kharagpur, India
{dheerendra,sourav}@maths.iitkgp.ernet.in

Abstract. Internet based content distribution presents a scalable platform for digital content trade to the remote users. It makes electronic commerce more profiting business. However, digital content can be easily copied and redistributed without any quality degradation over the network. Digital rights management (DRM) systems emerge as an effective solution which ensures copyright protection. Most of the existing DRM systems support only one way authentication where the server verifies user's authenticity and user simply assumed that he is interacting with the correct server. It may provide an opportunity of performing server spoofing attack to an adversary. In 2009, Zhang et al. presented a smart card based authentication scheme for DRM system in which user and server can mutually authenticate each other and establish a session key. Recently, Yang et al. demonstrated that Zhang et al.'s scheme is vulnerable to insider attack and stolen smart card attack. Additionally, they proposed an improved scheme to erase the drawbacks of Zhang et al.'s scheme. We identify that Yang et al.'s improved scheme is also vulnerable to password guessing attack and denial of service attack. Moreover, their scheme does not present efficient login and password change phases such that smart card executes the session in case of incorrect input. We show that how inefficiency of login and password change phases cause denial of service attack.

Keywords: Digital rights management, Smart card, Authentication.

1 Introduction

The advances in network technology have made internet an easy and efficient way for data transfer. The internet provides a scalable infrastructure for multimedia contents (music, movies, document, image, software, etc.) trade. It facilitates an easy access of multimedia content at low cost to the remote users. However, the content can be easily copied and redistributed over the network without degradation in the content quality. These drawbacks results rampant piracy, where piracy causes huge revenue to lose to the electronic commerce. Digital rights management (DRM) systems are developed in the response to the rapid increase in online piracy of commercially marketed multimedia products [3,2].

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 288–297, 2014.

Most of the schemes support one way authentication in which server verifies the user's authenticity and user simply assumes that he is interacting with the correct server. However, it provides the opportunity to the adversary to mislead the user by performing server impersonation attack. In 2009, Zhang et al. [5] presented a smart card based mutually authenticate and session key agreement scheme for DRM system in which user and server mutual authenticate each other and established a session key. Recently, Yang et al. [4] pointed out the flaws in Zhang et al.'s scheme. They showed that Zhang et al.'s scheme does not resist insider attack and stolen smart card attack. Furthermore, they proposed an improved scheme to eliminate all the drawbacks of Zhang et al.'s scheme. We analyze Yang et al.'s scheme and identify that their improved scheme is also vulnerable to password guessing attack and denial of service attack. Moreover, their scheme does not present efficient login and password change phases, where the inefficiency of incorrect input detection causes denial of service attack.

The rest of the paper is organized as follows: Section 2 presents the brief review of Yang et al.'s scheme. Section 3 points out the weakness of Yang et al.'s scheme. Finally, conclusion is drawn in Section 4.

2 Review of Yang et al.'s Smart Card Based Authentication Scheme for DRM

Yang et al. [4] presented an improved smart card based digital rights management authentication scheme to overcome the drawbacks of Zhang et al.'s scheme [5] scheme. Their scheme comprises the following participants: (i) Service Provider (SP); (ii) Terminal device (TD); (iii) Smart card (SC).

Yang et al.'s scheme provides mutual authentication among the service provider, terminal and smart-card, where the terminal is a playback device and smart-card stores user's sensitive information which is protected with a password. Their scheme comprises following three phases: (a) Registration; (b) Mutual authentication and key agreement; (c) Password update.

2.1 Registration

A user C registers to the server S and gets the smart card from the server as follows:

Step 1. C selects his password PW_C and a random nonce N_C, then computes $PWD = H_2(PW_C \oplus N_C)$. Then, C sends his identity ID_C with PWD to S via secure channel.

Step 2. Upon receiving registration request, S computes $K_C = H_2(sH_1(ID_C))$, and $S_C = K_C \oplus PWD$, where s denotes the server's secret key and, H_1 and H_2 denote the hash functions. Then, S embeds the parameters S_C into the smart card and issues the smart card to C via secure channel. It also stores $T_C = PWD \oplus K_C$ in its secure database.

Step 3. Upon receiving the smart card, C stores N_C in the smart card.

2.2 Mutual Authentication and Key Agreement

Step A_1. User → Device: $\{M_C\}$
To establish a session with server S, user C inserts his smart card into the card reader and inputs his identity ID_C and PW_C. Then the smart card performs the following steps:

1. It generates a random number r_c and a random nonce N'_C.
2. It computes $PWD' = H_2(PW_C \oplus N'_C)$, $M_{L1} = PWD' \oplus S_C$, $M_{L2} = H_2(S_C \oplus PWD')$ and $M_{(r_c)} = SKE.Enc_{K_C}(r_C)$.
3. It sends the message $M_C = \{SKE.Enc_{K_C}(r_c), M_{L1}, M_{L2}\}$ to the device.

Step A_2. Device → Server: $\{ID_v, ID_C, X, Y, M_C\}$
When the device receives the message from the smart card, it performs the following steps:

1. It generates a random number r_v and computes $X = r_v P$.
2. It computes $H_v = H_3(ID_v||ID_C||X||M_C)$, where H_3 denotes a hash function mapping an arbitrary length bit string into a random group member in Z_q^*.
3. It also computes $Y = r_v P_v + H_v S_v$ and sends the message $M_V = \{ID_v, ID_C, X, Y, M_C\}$ to the server.

Step A_3. Server → Device: $\{ID_s, ID_v, ID_C, M_{s1}, M_{s2}, M_{s3}, M_{s4}, M_{s5}, M_{s6}\}$
After receiving the device's message, the server performs the following tasks:

1. It computes $K_C = H_2(sH_1(ID_C))$ and $PWD = T_C \oplus K_C$ and $PWD' = M_{L1} \oplus K_C \oplus PWD$.
2. It verifies $M_{L2} =? H_2(K_C \oplus PWD \oplus PWD')$. If the verification does not hold, authentication breaks. Otherwise, the identity of the user is authenticated by the server. Then, server stores $T'_C = PWD' \oplus K_C$ into its database.
3. It computes $r'_c = SKE.Enc_{K_C}(SKE.Enc_{K_C}(r_c))$ and $H'_v = H_3 (ID_v||ID_C||X|| M_C)$.
4. It verifies $e(P, Y) =? e(P_v, X + H'_v P_s)$. If the verification does not succeed, the authentication fails. Otherwise, the identity of the device is authenticated by the server.
5. It generates two random strings r_{s1} and r_{s3}, and a random number r_{s2}.
6. It computes $M_{s1} = SKE.Enc_{K_C}(r_{s1})$, $M_{s2} = SKE.Enc_{s1}(H_2(ID_s||ID_v|| r_{s1}) \oplus r_c)$ and $M_{s3} = r_{s2}P$.
7. It computes the session key between the server and the device: $K_{sv} = H_2(r_v r_{s2} P)$ and between server and user: $K_{sc} = H_2(r_c||r_{s1})$.
8. It computes $M_{s5} = SKE.Enc_{K_{sv}(r_{s3})}$ and $M_{s6} = SKE.Enc_{K_{s1}(r_{s3})}$.
9. It also computes $M_{s4} = r_{s2}(P_S + P_C) + sh_s P$, where $h_s = H_3(ID_s||ID_v||ID_C||M_{s1}||M_{s2}||M_{s3}||M_{s5}||M_{s6})$.
10. The server sends the message $M_S = \{ID_s, ID_v, ID_C, M_{s1}, M_{s2}, M_{s3}, M_{s4}, M_{s5}, M_{s6}\}$ to the device.

Step A_4. Device → User: $\{ID_s, ID_v, M_{s1}, M_{s2}, M_{s6}, SKE.Enc_{r_{s3}}, r_{v1}\}$.

Upon receiving the server's message, the device executes the following steps:

1. It computes $P_S + P_C$ and $h'_s = H_3(ID_s||ID_v||ID_C||M_{s1}||M_{s2}||M_{s3}||M_{s5}|| M_{s6})$.
2. It verifies $e(P, M_{s4}) =?\ e(P_S + P_C, M_{s3})e(h'_s P, sP)$. If the verification does not succeed, the authentication fails. Otherwise, the identity of the server is authenticated by the device.
3. It computes the session key between the server and the device $K_{sv} = H_2(r_v M_{s3})$.
4. It computes $r'_{s3} = D_{sv}(M_{s5})$ and generates a random string r_{v1}.
5. It also computes $SKE.Enc_{r_{s3'}}(r_{v1})$, then sends the message $\{ID_s, ID_v, M_{s1}, M_{s2}, M_{s6}, SKE.Enc_{r_{s3}}, r_{v1}\}$ to the smart card.

Step A_5. User → Device: $\{SKE.Enc_{s3}(r_{u1})\}$.

Upon receiving the message, the smart card executes the following steps:

1. It computes $r'_{s1} = D_{K_C}(M_{s1})$ and verifies $D_{s1'}(M_{s2}) \oplus r_c = ?\ H_2(ID_s||ID_v||r'_{s1})$. If the verification succeeds, the server is authenticated by the smart card. The smart card replaces stored values N_C and S_C with N'_C and $S'_C = S_C \oplus PWD \oplus PWD'$, respectively.
2. It computes the session key $K_{sc} = H_2(r_c||r_{s1})$ between the server and the user.
3. It decrypts $r_{s3} = D_{s1}(M_{s6})$ and Computes $r_{v1} = D_{r_{s3}}(SKE.Enc_{r_{s3}}(r_{v1}))$.
4. it generates a random string r_{u1} and sends $SKE.Enc_{r_{s3}}(r_{u1})$ back to the device.
5. It computes the session key $K_{uv} = H_2(r_{u1||r_{v1}})$ between the user and the device.
6. Upon receiving the message from the smart card, the device decrypts
$r_{u1} = D_{r_{s3}}(SKE.Enc_{r_{s3}}(r_{u1}))$ and computes the session key $K_{uv} = H_2(r_{u1}||r_{v1})$ with the user.

2.3 Password Update Phase

When the user wishes to change the password of smart card, he inserts his smart card and inputs the identity ID_C, the old password PW_C and the new password PW_{new}. Then, smart card executes the following steps:

Step P_1. User → Server: $\{m_c\}$

1. It generates a random nonce N'_C and computes $PWD' = H_2(PW_C \oplus N'_C)$.
2. It also computes the following values:

$$\begin{aligned} M_{c1} &= PWD' \oplus S_C \\ PWD_{new} &= H_2(PW_{new} \oplus N'_C) \\ M_{c2} &= H_2(S_C \oplus PWD' \oplus PWD_{new}) \\ M_{c3} &= PWD_{new} \oplus S_C \end{aligned}$$

3. Then, the smart card sends the message $M_c = \{M_{c1}, M_{c2}, M_{c3}\}$ to the server.

Step P_2. Server $\rightarrow$ User : $\{m_{c4}\}$

Upon receiving the message, the server executes the following steps:

1. It computes the following parameters:

$$\begin{aligned} K_C &= H_2(sh_1(ID_C)) \\ PWD &= T_C \oplus K_C \\ PWD' &= M_{c1} \oplus K_C \oplus PWD \\ PWD_{new} &= M_{c3} \oplus K_C \oplus PWD \end{aligned}$$

2. It verifies $M_{c2} =? H_2(K_C \oplus PWD \oplus PWD' \oplus PWD_{new})$. If the verification succeeds, the server accepts the request and updates T_C with $T'_C = PWD_{new} \oplus K_C$.
3. Finally, the server sends $M_{c4} = H_2(PWD_{new} \oplus K_C \oplus PWD)$ back to the user.

Step P_3. Upon receiving the message, the smart card verifies $M_{c4} =? H_2(PWD_{new} \oplus S_C)$. If the verification succeeds, the server is authenticated. Then, the smart card replaces N_C and S with N'_C and S'_C, respectively, where $S'_C = S_C \oplus PWD \oplus PWD_{new}$.

3 Cryptanalysis of Yang et al.'s Scheme

In this section, we demonstrate the attacks on Yang et al.'s scheme [4].

3.1 Off-Line Password Guessing Attack

In general, user selects a password, which he can easily remember, as he has to use his password every time during login. It gives the opportunity of password guessing to an adversary (E). Instead of this fact, a smart card based protocol should be able to resist password guessing attack. We analyze Yang et al.'s scheme and find out that their scheme does not resist password guessing attack, which can be justified as follows:

Step G_1. E can retrieve the secrets N_C and S_C from the stolen smart card.

Step G_2. E can compute the values $M_{(r_c)} = SKE.Enc_{K_C}(r_C)$ from M_V and, $M_{s1} = SKE.Enc_{K_C}(r_{s1})$ and $M_{s2} = SKE.Enc_{s1}(H_2(ID_s||ID_v||r_{s1}) \oplus r_c)$ from M_S, as M_V and M_S transmit via public channel, and an adversary can intercept and record the message transmitting via public channel.

Step G_3. E guesses the password PW_C^* and computes $K_C^* = S_C \oplus h(PW_C^* \oplus N_C)$ and $r_C^* = SKE.Dec_{K_C^*}(SKE.Enc_{K_C}(r_c))$. E also computes $r_{s1}^* = SKE.Dec_{K_C^*}\ (M_{s1})$, then verifies $M_{s2} =? SKE.Enc_{r_{s1}^*}(H_2(ID_s||ID_v||r_{s1}^*) \oplus r_c^*)$.

Step G_4. If the verification succeeds, E identifies PW_C^* as the user's password. Otherwise, E repeats *Step* G_3 until succeeded.

3.2 Denial of Service Attack

An adversary can successfully perform denial of service attack such that a legitimate user cannot establish authorized session with the server, *i.e.*, the user can not login to the server once the denial of service attack succeeds. An adversary (E) can perform denial of service attack in Yang et al.'s scheme as follows:

- When a device sends the message M_V to the server S, the server S verifies the authenticity of the message M_V. When the verification holds, the server updates $T_C(= PWD \oplus K_C)$ with $T'_C(= PWD' \oplus K_C)$ and responds with the message $M_S = \{ID_s, ID_v, ID_C, M_{s1}, M_{s2}, M_{s3}, M_{s4}, M_{s5}, M_{s6}\}$.
- E intercepts the message M_S and replaces M_S with M'_S, where $M^*_S = \{ID_s, ID_v, ID_C, M_{s1}, M^*_{s2}, M_{s3}, M_{s4}, M_{s5}, M_{s6}\}$. E computes M^*_{s2} as follows:
  - Select a random value r_E.
  - Compute $M^*_{s2} = M_{s2} \oplus r_E$, *i.e.*, $M^*_{s2} = SKE.Enc_{s1}(H_2(ID_s||ID_v||r_{s1}) \oplus r_c) \oplus r_E$.
- Upon receiving the message M^*_S, device computes $h^*_s = H_3(ID_s||ID_v||ID_C|| M_{s1}||M^*_{s2}|| M_{s3}||M_{s5}||M_{s6})$, which is not equal to h_s, as $M_{s2} \neq M^*_{s2}$.
- When the device verifies the condition $e(P, M_{s4}) \ =?\ e(P_S + P_C, M_{s3})e(h^*_s P, sP)$. The verification does not hold as $h^*_s \neq h_s$. The authentication fails. Moreover, when the smart card computes $r'_{s1} = D_{K_C}(M_{s1})$ and verifies $D_{s1'}(M^*_{s2}) \oplus r_c =?\ h_2(ID_s||ID_v||r'_{s1})$. The authentication does not hold, as $M^*_{s2} \neq M_{s2}$.
- Since, the authentication fails, the smart card does not replace the values N_C and S_C with N'_C and $S'_C = S_C \oplus PWD \oplus PWD_{new}$, respectively, where $PWD = H_2(PW_C \oplus N_C)$ and $PWD' = H_2(PW_C \oplus N'_C)$.

It is clear from the discussion that if the verification fails at the user's end, *i.e.*, the user message authentication holds but the server message authentication does not hold, the smart card does not replace the stored values N_C and S_C as server message authentication does not hold. However, the server updates the value T_C with T'_C as user message authentication holds. In other words, if the verification of the user's message holds but the servers's message does not hold, the server updates its parameters but the smart card does not update its parameters. It causes denial of server attack and a user cannot establish authorized session with a server, once the denial of service attack succeeds. It is clear from the following steps:

- When the user initiates the session, the smart card generates a random number r_c and a random nonce N''_C, then computes $PWD'' = H_2(PW_C \oplus N''_C)$, $M'_{L1} = PWD'' \oplus S_C$, $M'_{L2} = H_2(S_C \oplus PWD'')$ and $M_{(r_c)} = SKE.Enc_{K_C}(r_C)$. The smart card sends the message $M'_C = \{SKE.Enc_{K_C}(r_c), M'_{L1}, M'_{L2}\}$ to the device.
- Upon receiving the message M'_C, the device computes the messages $M'_V = \{ID_v, ID_C,\ X = r_v P, Y = r_v P_v + H_v S_v, M'_C\}$ for random number r_v and sends it to the server, as described in *Step* A_2.

- Upon receiving the message M'_V, the server computes $K_C = H_2(sH_1(ID_C))$ and $PWD' = T'_C \oplus K_C$ where $PWD' = h(PW_C \oplus N'_C)$.
- The server also computes $PWD''' = M_{L1} \oplus K_C \oplus PWD' = PWD'' \oplus S_C \oplus K_C \oplus PWD' = PWD'' \oplus PWD \oplus PWD' \neq PWD$, as $PWD' \neq PWD''$. $T'_C = PWD' \oplus K_C \ h(PW^*_C \oplus N_C)$
- The server verifies $M_{L2} =?\ H_2(K_C \oplus PWD \oplus PWD''')$. The verification does not hold, as $PWD''' \neq PWD''$, the session is terminated.

The above discussion shows that user cannot establish an authorized session t the server with the current smart card. It proves that Yang et al.'s scheme does not resist denial of service attack.

3.3 Inefficient Login Phase

An efficient smart card based remote user authentication scheme should be able to identify incorrect login request so that extra communication and computation cost should not occur due to incorrect login phase [1]. However, in Yang et al.'s scheme, the smart card does not verify the correctness of input and executes the login session in case of incorrect input which causes a denial of service attack.

- The user inputs identity ID_C and incorrect password PW^*_C by mistake, *i.e.*, $PW^*_C \neq PW^*_C$.
- Upon receiving the input, the smart card executes the session without verifying the correctness of inputs as follows:
  - Generate a random number r_c and a random nonce N'_C.
  - Compute $PWD'^* = H_2(PW^*_C \oplus N'_C)$, $M_{L1} = PWD'^* \oplus S_C$ and $M_{L2} = H_2(S_C \oplus PWD'^*)$.
  - Compute $PWD^* = H_2(PW^*_C \oplus N_C)$ and get $K^*_C = S_C \oplus PWD^* \neq K_C$, as $PWD^* \neq PWD$.
  - Compute $SKE.Enc_{K^*_C}(r_c)$ and send the message $M^*_C = \{SKE.Enc_{K^*_C}(r_c), M_{L1}, M_{L2}\}$ to the device.
- Upon receiving the smart card's message, device computes the message $\{ID_v, ID_C, X, Y, M^*_C\}$ and sends it to the server.
- upon receiving the device message, the server computes $K_C = H_2(sH_1(ID_C))$, $PWD = T_C \oplus K_C$ and $PWD'^* = M_{L1} \oplus K_C \oplus PWD$, then verifies $M_{L2} =?\ H_2(K_C \oplus PWD \oplus PWD'^*)$. The verification holds, as $M_{L2} = H_2(S_C \oplus PWD'^*) = H_2(K_C \oplus PWD \oplus PWD'^*)$. Since, the verification succeeds, the server stores $T'^*_C = PWD'^* \oplus K_C$ into its database. Then, the server computes the following values:
  - Compute $r^*_c = SKE.Dec_{K_C}(SKE.Enc_{K^*_C}(r_c)) \neq r_C$, as $K_C \neq K^*_C$.
  - Compute $H'_v = H_3(ID_v||ID_C||X||M_C)$ and verify $e(P, Y) =?\ e(P_v, X + H'_v P_s)$. When the verification holds, identity of the device is authenticated.
  - Generate and send the message $M_S = \{ID_s, ID_v, ID_C, M_{s1}, M_{s2}, M_{s3}, M_{s4}, M_{s5}, M_{s6}\}$ to a device which is similar to the *Step* A_3 of authentication phase.

- Device verifies the condition $e(P, M_{s4}) =?\ e(P_S + P_C, M_{s3})e(h'_s P, sP)$. The authentication holds and the server sends the message $\{ID_s, ID_v, M_{s1}, M_{s2}, M_{s6},\ SKE.Enc_{r_{s3}}, r_{v1}\}$ to user as discussed in *Step* A_5 of authentication phase. Then, device executes the following steps:
    1. Device computes $P_S + P_C$ and $h'_s = H_3(ID_s||ID_v||ID_C||M_{s1}||M_{s2}||M_{s3}||M_{s5}||M_{s6})$.
    2. Device verifies $e(P, M_{s4}) =?\ e(P_S + P_C, M_{s3})e(h'_s P, sP)$. If the verification does not succeed, the authentication fails. Otherwise, the identity of the server is authenticated by the device.
    3. Device computes the session key between the server and the device $K_{sv} = H_2(r_v M_{s3})$.
    4. It computes $r'_{s3} = D_{sv}(M_{s5})$ and generates a random string r_{v1}.
    5. Device also computes $SKE.Enc_{r_{s3'}}(r_{v1})$, then sends the message $\{ID_s, ID_v, M_{s1}, M_{s2}, M_{s6}, SKE.Enc_{r_{s3}}, r_{v1}\}$ to the smart card.
- Upon receiving the message, the smart card computes $r^*_{s1} = SKE.Dec_{K^*_C}(M_{s1}) \neq r_{s1}$, as $K^*_C \neq K_C$.
- The smart card verifies $SKE.Dec_{r^*_{s1}}(M_{s2}) \oplus r_c =?\ H_2(ID_s||ID_v||r^*_{s1})$. The verification does not hold, as $r^*_{s1} \neq r_{s1}$. Then, smart card terminates the session and does not replace stored values N_C and S_C with N'_C and $S'_C = S_C \oplus PWD \oplus PWD'^*$, respectively.

It is clear from the above discussion that in case of wrong password input, the authentication at server end succeeds but user end fails. In other words, the server updates the T'_C with T_C, although the user does not update due to authentication failure. It causes **denial of service attack** as discussed in section 3.2.

3.4 Inefficient Password Change Phase

In Yang et al.'s scheme, the smart card does not verify the correctness of input password and executes the password change request in case of wrong input. If a user inputs wrong password PW^*_C instead of PW_C, then the smart card executes the password change phase as follows:.

- Upon receiving the input ID_C and PW^*_C, the smart card does not verify the correctness of input and generates a random nonce N'_C. Smart card computes $PWD'^* = H_2(PW^*_C \oplus N'_C)$. Note that $PWD'^* \neq PWD'(= H_2(PW_C \oplus N'_C)$ as $PW^*_C \neq PW_C$.
- Smart card computes $M^*_{c1} = PWD'^* \oplus S_C, PWD_{new} = H_2(PW_{new} \oplus N'_C), M^*_{c2} = H_2(S_C \oplus PWD'^* \oplus PWD_{new})$ and $M_{c3} = PWD_{new} \oplus S_C$. Then, the smart card sends the message $M^*_c = \{M^*_{c1}, M^*_{c2}, M_{c3}\}$ to the server.
- Upon receiving the message M_c, the server computes $K_C = H_2(sh_1(ID_C))$, $PWD = T_C \oplus K_C$, $PWD'^* = M^*_{c1} \oplus K_C \oplus PWD$ and $PWD_{new} = M_{c3} \oplus K_C \oplus PWD$

- Server verifies $M_{c2}^{*} =? H_2(K_C \oplus PWD \oplus PWD'^{*} \oplus PWD_{new})$. The verification succeeds as $M_{c2}^{*} = H_2(S_C \oplus PWD'^{*} \oplus PWD_{new}) = H_2(K_C \oplus PWD \oplus PWD'^{*} \oplus PWD_{new})$. Then, the server accepts the request and updates T_C with $T'_C = PWD_{new} \oplus K_C$. The server also sends the message $M_{c4} = H_2(K_U \oplus PWD \oplus PWD_{new})$ to the user.
- Upon receiving the message, the smart card verifies $M_{c4} =? H_2(PWD_{new} \oplus S_C)$. The verification succeeds as $M_{c4} = H_2(K_U \oplus PWD \oplus PWD_{new}) = H_2(PWD_{new} \oplus S_C)$. Then, the smart card replaces N_C and S_C with N'_C and S'_C, respectively, where $S_C'^{*} = S_C \oplus PWD^{*} \oplus PWD_{new}$. Note that $S_C'^{*} \neq K_C \oplus PWD_{new}$, because of the following facts:

$$\begin{aligned} S_C'^{*} &= S_C \oplus PWD^{*} \oplus PWD_{new} \\ &= K_C \oplus PWD \oplus PWD^{*} \oplus PWD_{new} \\ &\neq K_C \oplus PWD_{new} \texttt{as } PWD(= PW_C \oplus N_C) \neq PWD^{*}(= PW_C^{*} \oplus N_C). \end{aligned}$$

In case of wrong password input, the password change phase also succeeds and user and server update their parameters. The server updates the information T_C with $T'_C(= PWD_{new} \oplus K_C)$ correctly. However, the user updates S_C with $S_C'^{*}(\neq K_C \oplus PWD_{new})$. It shows that user does not update his parameters correctly in case of wrong input. It causes **denial of service attack** which is justified as follows:

- After the password update, user inputs his identity ID_C and new password PW_{new} to login to the server.
- Upon receiving the input, the smart card generates a random number r_c and a random nonce N''_C, and computes $PWD'_{new} = H_2(PW_{new} \oplus N''_C)$, $M_{L1}^{*} = PWD'_{new} \oplus S_C'^{*}$, $M_{L2}^{*} = H_2(S_C'^{*} \oplus PWD'_{new})$.
- The smart card computes $K_C^{*} = S_C'^{*} \oplus PWD_{new} = S_C \oplus PWD^{*} \oplus PWD_{new} \oplus PWD_{new} = K_C \oplus PWD \oplus PWD^{*} \neq K_C$, as $PWD \neq PWD^{*}$. It also computes $M_{(r_c)}^{*} = SKE.Enc_{K_C^{*}}(r_C)$. Then, it sends the message $M_C^{*} = \{SKE.Enc_{K_C^{*}}(r_c), M_{L1}^{*}, M_{L2}^{*}\}$ to the device.
- Upon receiving the message M_C^{*}, the device computes the messages $M_V^{*} = \{ID_v, ID_C, X = r_vP, Y = r_vP_v + H_vS_v, M_C^{*}\}$ for random number r_v and sends it to the server, as described in *Step* A_2.
- Upon receiving the message M_V^{*}, the server computes $K_C = H_2(sH_1(ID_C))$ and $PWD_{new} = T'_C \oplus K_C$.
- The server also computes $PWD'^{*}_{new} = M_{L1}^{*} \oplus K_C \oplus PWD_{new} = PWD'_{new} \oplus S_C'^{*} \oplus K_C \oplus PWD_{new} = PWD'_{new} \oplus K_C \oplus PWD \oplus PWD^{*} \oplus PWD_{new} \oplus K_C \oplus PWD_{new} = PWD'_{new} \oplus PWD \oplus PWD^{*} \neq PWD'_{new}$, as $PWD \neq PWD^{*}$. $T'_C = PWD' \oplus K_C \; h(PW_C^{*} \oplus N_C)$
- The server verifies $M_{L2}^{*} =? H_2(K_C \oplus PWD \oplus PWD'^{*}_{new})$. The verification does not hold as $M_{L2}^{*} = H_2(S_C'^{*} \oplus PWD'_{new}) = H_2(K_C \oplus PWD \oplus PWD^{*} \oplus PWD_{new} \oplus PWD'_{new})$ and $PWD \neq PWD^{*}$. The session is terminated as verification does not hold.

The above discussion shows that user cannot establish an authorized session with the server. It shows that Yang et al.'s scheme fails to provide efficient password change phase.

4 Conclusion and Future Scope

We have presented a study on Yang et al.'s scheme which shows that their scheme is vulnerable to off-line password guessing attack and denial of service attack. Moreover, the flaws in the login and password change phases cause denial of service attack. The study clearly demonstrates that Yang et al.'s scheme fails to satisfy desirable security attribute which an efficient smart card based authentication scheme should satisfies to ensure secure communication over the public channel. In other words, security of Yang et al.'s scheme is completely broken. This indicates that an improved authenticated key agreement scheme is required for DRM system which can resist all the attack where Yang et al.'s scheme fails and can present efficient login and password change phases.

Acknowledgement. First author is grateful to University Grants Commission (UGC), New Delhi, India, for their financial support.

References

1. Mishra, D.: A study on id-based authentication schemes for telecare medical information system. arXiv preprint arXiv:1311.0151 (2013)
2. Mishra, D., Mukhopadhyay, S.: Secure content delivery in drm system with consumer privacy. In: Deng, R.H., Feng, T. (eds.) ISPEC 2013. LNCS, vol. 7863, pp. 321–335. Springer, Heidelberg (2013)
3. Subramanya, S., Yi, B.K.: Digital rights management. IEEE Potentials 25(2), 31–34 (2006)
4. Yang, H.W., Yang, C.C., Lin, W.: Enhanced digital rights management authentication scheme based on smart card. Institution of Engineering and Technology (2013)
5. Zhang, Y.C., Yang, L., Xu, P., Zhan, Y.S.: A drm authentication scheme based on smart-card. In: International Conference on Computational Intelligence and Security, CIS 2009, vol. 2, pp. 202–207. IEEE (2009)

Automatic Evaluation and Signature Generation Technique for Thwarting Zero-Day Attacks

Ratinder Kaur and Maninder Singh

Computer Science and Engineering Department,
Thapar University, Patiala-147004, India

Abstract. Zero-day attack is a cyber-attack which exploits vulnerabilities that have not been disclosed publicly. Zero-day attacks are very expensive and powerful attack tools. They are used in conjunction with highly sophisticated and targeted attacks to achieve stealthiness with respect to standard intrusion detection techniques. Zero-day attacks are unknown and have no signature so they are difficult to detect. This paper presents a novel and efficient technique for detecting zero-day attacks. The proposed technique detects obfuscated zero-day attacks with two-level evaluation, generates signature for new attack automatically and updates other sensors by using push technology via global hotfix feature.

Keywords: Zero-Day Attack, Honeynet, Obfuscation, Signature Generation, Push Technology.

1 Introduction

News of zero-day attacks dominates the headlines. People talked about zero-day attacks few years back, but today every industry faces it. The modern zero-day attacks not only deals with the freshness of the vulnerability it can exploit but can also exhibit any number of distinct behaviors. This includes: complex mutation to evade defenses, multi-vulnerability scanning to identify potential targets, targeted exploitation that launches directed attacks against vulnerable hosts, remote shells that open arbitrary ports on compromised hosts to connect to at a later time, malware drops, in which malicious code is downloaded from an external source to continue propagation. Zero-day attacks occur during the vulnerability window that exists in the time between when vulnerability is first exploited and when software developers start to develop a counter to that threat. According to an empirical study, a typical zero-day attack may last for 312 days on average [1] [17].

In recent years the number of zero-day attacks reported each year has increased immensely. According to Symantec's Internet Security Threat Report of 2013 [2] there is 42% increase in zero-day targeted attacks in 2012. The most dangerous zero-day exploits ever seen in cyberspace are Hydraq trojan, Stuxnet, Duqu and Flame. The Hydraq trojan, also known as Aurora attack aimed to steal information from several companies. Stuxnet worm (discovered in June 2010) an

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 298–309, 2014.

incapacitating computer virus that wiped out nearly 60% of Iran's computer network. Duqu (discovered in September 2011) related to Stuxnet worm exploits zero-day Windows kernel vulnerabilities, uses stolen digital keys and is highly targeted. Flame (discovered in 2012) is a modular computer malware that exploits some same zero-day vulnerabilities in Microsoft Windows as Stuxnet.

In this paper an efficient novel technique for detecting zero-day attacks is proposed. It detects obfuscated zero-day attacks with two-level evaluation (First-level: *Detects Unknown* and Second-level: *Confirms Malicious*) and generates new signatures automatically to update other IDS/IPS sensors via global hotfix. The remainder of the paper is organized as follows. In section 2, related work is summarized. In section 3, detailed working of the proposed technique is presented. Finally in section 4, experimental evaluation is described with results and paper is concluded.

2 Related Work

To defend against zero-day attacks, the research community has proposed various techniques. These techniques can be broadly classified into: statistical-based, signature-based, behavior-based and hybrid techniques.

2.1 Statistical-Based

The statistical-based techniques uses various statistical methods like distributed Sequential Hypothesis Testing (dSHT) [4], One-class Support Vector Machine (SVM) with Sequential Minimal Optimization (SMO) [5], Rough Set Theory (RST) [6], Principal Component Analysis (PCA) [7], Supervised Learning [19], Contextual Anomaly Detection [20][24], Combined Supervised and Unsupervised Learning [23] to detect zero-day polymorphic worms. These techniques are independent of worm signatures and thus, take a different approach to detect zero-day worms. These techniques are dependent on attack profiles build from historical data. Due to the static nature of attack profiles, the detection techniques are unable to adapt to the timely changes in the environment. Setting the limit (or detection parameters) for judging new observations is a critical step. If the threshold value is very narrow, it will frequently be exceeded resulting in a high rate of false positive alarms, and if it is very wide the limit will never be exceeded, resulting in many false negative alarms. In statistical-based detection the detection parameters are either manually extracted or adjusted to detect new attacks. All these factors, limit the statistical detection approaches to work in off-line mode. And hence, they cannot be used for instant detection and protection in real time.

2.2 Signature-Based

The signature-based detection techniques mainly focus on polymorphic worms. Several polymorphic worm signature generation schemes are surveyed. Based on their characteristics, the signatures are classified into four categories:-

Content-Based. Content-based detection [8][9][16][25] relies on using byte pattern-based worm signatures to detect worm traffic. When the byte pattern of a given traffic flow matches the byte pattern defined by a worm signature, that traffic is identified as being worm traffic. In order to create these signatures, systems have been proposed to look for common byte stream patterns. These signatures capture the features specific to a worm implementation, thus might not be generic enough and can be evaded by other exploits as there can be worms which have no content-based signature at all [3]. Furthermore, various attacks have been proposed to evade the content-based signatures by misleading signature generation processes using crafted packets injection into normal traffic.

Flexible Content-Based. The systems that generates flexible content-based signatures [22][10] work on a byte level and are flexible in the way that they do not just try to match strings or substrings with incoming packets but their signatures describe patterns of how malicious bytes are organized. Example techniques use byte-frequency distributions (instead of fixed values) or regular expressions. Such signatures are difficult to transform into Snort signatures and hence, cannot be deployed widely for online detection.

Semantic-Based. These approaches go beyond the byte level examination. Instead of using repeated substring found in the network stream they either use structure of the executable code present in the network stream or attack analysis information [11] to generate semantic signatures. To identify the semantics-derived characteristics of worm payloads, existing techniques detect the invariant characteristics reflecting semantics of malicious codes (e.g., behavioral characteristics of the decryption routine of a polymorphic worm). These signatures are robust to evasion attempts because it considers more about semantics. However, the semantics analysis introduce non-trivial performance overheads and are computationally expensive to generate as compared to approaches based on substrings.

Vulnerability-Driven. Vulnerability-driven signature captures [12] the characteristics of the vulnerability the worm exploits. They analyze vulnerabilities in a program, its execution, and conditions needed to exploit that vulnerability. The signatures based on the vulnerability typically doesn't change so they are robust against exploits that have variances and can morph. These signatures only require intimate knowledge of the vulnerabilities and can be developed prior to any known exploits, allowing them to be proactive. Unfortunately, vulnerability-driven signatures are difficult to generate and due to the increased vagueness of the signature, this method can also lead to more false-positives. Moreover, the existing vulnerability-driven schemes are mostly host-based, and some suffers from computational overhead and are specific to buffer-overflow attacks only. These techniques may be inefficient if they are not directly based on the exact vulnerability analysis and lack vulnerability details.

2.3 Behavior-Based

Behavior-based techniques [13][18][14] look for the essential characteristics (indicators) of worms which do not require the examination of payload byte patterns. They focus on the actual dynamics of the worm execution to detect them. They monitor the dynamic behavior of malicious activity rather than its static characteristics. The executing processes are monitored to analyze their behavior in a controlled simulated environment. It is an effective way to detect new threats. This approach may be dangerous to rely on because the malware might cause harm to the system before it is recognized as malicious. Even if a malicious program is scanned by observing its execution in a virtual environment, this technique may not effectively capture the context in which the malicious program interacts with the real victim machine. Behavior-based techniques may detect a wide range of novel attacks with low false positives but they can be easily evaded once the behavioral analysis method is known.

2.4 Hybrid Techniques

HDPS is a hybrid technique that employs a heuristic approach to detect return address and to filter mass innocent network flows [15]. It applies a sliding window to detect return address. It uses Markov Model to detect the existence and location of executable codes in suspicious flows and applies a sliding window to identify the executable code. And finally, it applies an elaborate approach to detect NOP (no-operation) sleds. Honeyfarm [21] is another hybrid scheme that combines anomaly and signature detection with honeypots. The system works at three levels. At first level signature based detection is used to filter known worm attacks. At second level an anomaly detector is set up to detect any deviation from the normal behavior. In the last level honeypots are deployed to detect zero day attacks. Low interaction honeypots are used to track attacker activities while high interaction honeypots help in analyzing new attacks and vulnerabilities.

3 Proposed Technique

3.1 Architecture

An efficient two-level evaluation technique is proposed to minimize the impact of above identified challenges during zero day attack detection. The basic operation of the proposed technique is described by the flowchart in Figure 1. The network traffic is captured and filtered for known attacks. If the filtered traffic is found suspicious of containing some unknown attack (captured by the Honeynet and undetected by sensor), that unknown traffic trace is evaluated for zero-day attack. If the traffic trace is found benign after evaluation the whitelist in sensors is updated. But if it's found malicious then a new signature is generated and pushed to local and remote sensors through global hotfix for containing the zero-day attack.

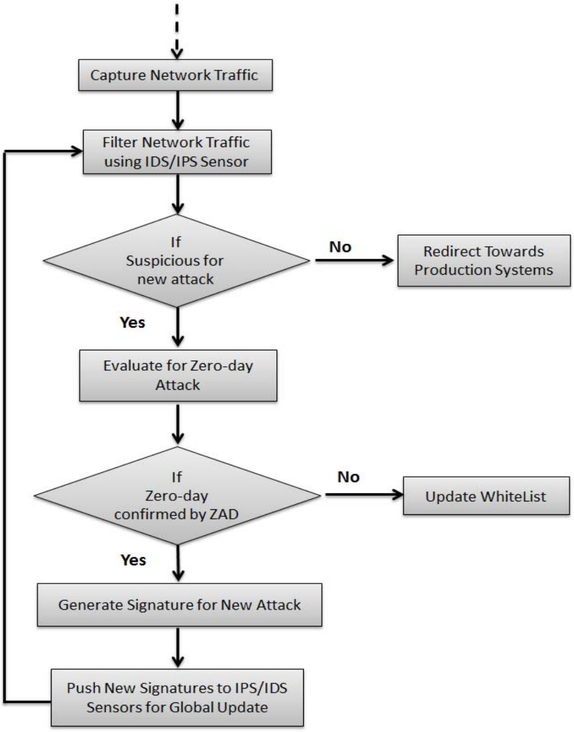

Capture Network Traffic
Filter Network Traffic using IDS/IPS Sensor
If Suspicious for new attack
No
Redirect Towards Production Systems
Yes
Evaluate for Zero-day Attack
If Zero-day confirmed by ZAD
No
Update WhiteList
Yes
Generate Signature for New Attack
Push New Signatures to IPS/IDS Sensors for Global Update

Fig. 1. System Flowchart

Figure 2 shows the basic architecture of our proposed technique. It comprises different components - Port Mirroring Switch, Honeynet, Intrusion Detection and Prevention (IDS/IPS) Sensors, Zero-day Attack Detection (ZAD) System and Global IDS/IPS Hotfix Server. Router connects the entire setup to the Internet. Port mirroring switch passes network traffic simultaneously to both Honeynet and IDS/IPS sensors. Honeynet is used to identify the mechanism of a new attack and to collect evidence for attacker's activity. When a new attack is encountered the network traffic associated with that attack is logged and is redirected to the high-interaction honeypots. The honeypots interact with the attacker and the entire communication is logged. The network logs and honeypot system interaction logs collectively addressed as “Honeynet Trace” or “Unknown Attack Trace” are kept for further analysis.

The IDS/IPS sensor filters known attacks for the same traffic and stores rest of the filtered traffic in an online repository. Then the data collected from both honeynet and IDS/IPS sensor is compared and analyzed in ZAD. The ZAD system examines if similar unknown attack traces are found in IDS/IPS sensor's filtered traffic or not. If similar attack traces are found, then that is a candidate for zero-day attack undetected by IDS/IPS sensor. Up to this level, this is assured that there is some malicious traffic which was missed by sensors. This could only

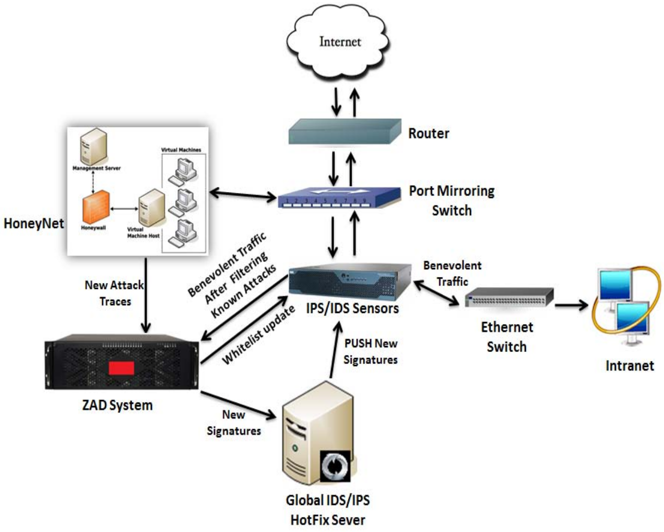


Fig. 2. Architecture of Proposed Model

happen when the IDS/IPS sensor does not have matching signature for the unknown malicious traffic in its database. After finding the candidate for zero-day attack it is necessary to do further analysis to confirm its malicious intent and to generate a new signature for it. Next level evaluation for zero-day attack is discussed in the following section.

3.2 Evaluating Zero-Day Attack

The candidate for zero-day attack may result in false positive so it's essential to evaluate it. The evaluation process is used to confirm the malicious intentions of the candidate. This evaluation is done by ZAD-Analyzer in ZAD system. The internal process flow of ZAD system is shown in Figure 3.

ZAD takes input from both Honeynet and IDS/IPS sensors to compare and extract the zero-day attack candidate. For comparing, ZAD uses an efficient algorithm known as Longest Common Prefix (LCP). LCP constructs a suffix tree to match attack trace. The attack trace is then input to an Emulator for per byte execution. Emulator is the right choice for analyzing decrypted and obfuscated code. The attack trace is executed in the emulation environment and is allowed to perform malicious activities. After execution the system anomalies are analyzed. The system anomalies help to determine a system's status (whether or not malicious code is present) by comparing the system status information to a standard. For this, the abstract method of analyzing system anomalies is used that is validating checksums of critical files.

Our analysis is based on the fact that it is not possible to compromise a system without altering a system file. A malicious code can only do one of

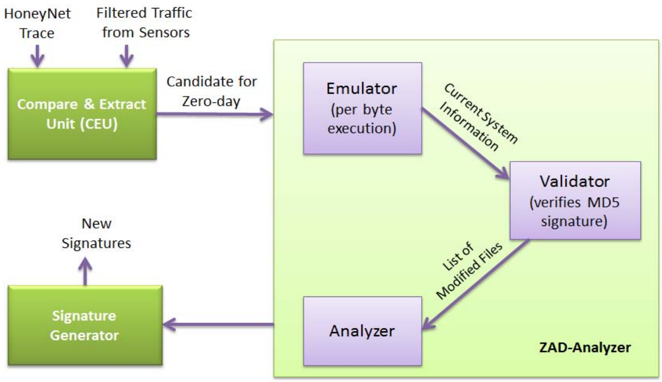

HoneyNet Trace
Filtered Traffic from Sensors
Compare & Extract Unit (CEU)
Candidate for Zero-day
Emulator (per byte execution)
Current System Information
Validator (verifies MD5 signature)
List of Modified Files
Analyzer
ZAD-Analyzer
New Signatures
Signature Generator

Fig. 3. ZAD-Analyzer Internal Process Flow

three things: add, remove or modify files. It can remove system logs. It can add tools such sniffers or viruses for later use. And most important it can change the system in numerous ways like new accounts, modified passwords, tweaked registries, trojaned files etc. To analyze such changes, a Validator is used in ZAD-Analyzer. Validator maintains a MD5 checksum database of original files of the Emulator Operating System. After the execution of attack trace in Emulator, the file system gets corrupted. The Emulator then sends current file system status to the Validator. The Validator then recalculate MD5 checksum on all files of Emulator Operating System and compares with baseline MD5 checksum database. The result of this comparison is “List of Modified Files” which is sent to the Analyzer. The Analyzer unit then crosschecks the “List of Modified Files” with the “List of Critical Files” maintained. Critical files for eg. in Windows can be registry files, startup files, system configuration files, password files etc. If the critical files are modified, it proves that the candidate is a real zero-day attack.

Thus, the system does two-level evaluation for detecting zero-day attack. First-level (*Detects UnKnown*) where Honeynet flags a new suspicious event and IDS/IPS sensors ignores it. Second-level (*Confirms Malicious*) where MD5 baseline is used to confirm its malicious intentions. This two-level (*Detects Unknown Malicious*) evaluation decreases the false positives to nearly zero. After confirming a zero-day attack, ZAD-Analyzer commands the Signature Generator to generate signature for new attack. On the other hand, if no critical file is changed then the candidate is false positive and the Whitelist is updated. The corrupted Emulator Operating System is replaced with fresh new installation to again start the execution of other attack trace.

3.3 Signature Generation and Update

After evaluation zero-day attack packets are fed to next module for signature generation. This module generates a common token-subsequence signature for a set of attack packets by applying the Longest Common Subsequence (LCSeq) algorithm. The algorithm compares two zero-day attack packets to get the longest common subsequence between them. Let two sequences be defined as follows: X = (x1, x2...xm) and Y = (y1, y2...yn). Let LCSeq(Xi, Yj) represent the set of longest common subsequence of prefixes Xi and Yj. This set of sequences is given by the following.

$$LCSeq(Xi, Yj) = \begin{cases} \Phi & \text{if i=0 or j=0} \\ LCSeq(Xi-1, Yj-1)+1 & \text{if xi = yj} \\ longest(LCSeq(Xi, Yj-1), LCSeq(Xi-1, Yj)) & \text{if xi} \neq \text{yj} \end{cases}$$

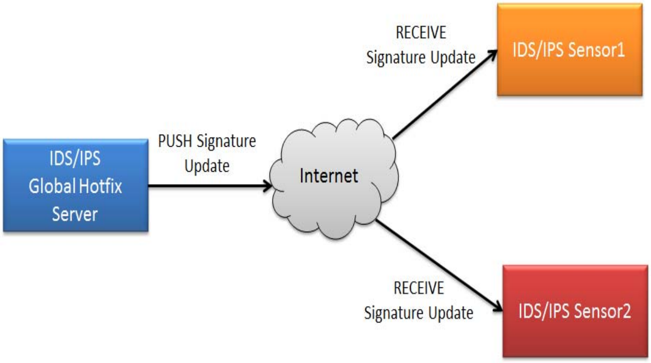


Fig. 4. Signature Update by Global Hotfix

The new attack signatures generated by ZAD are sent to a server responsible for global IDS/IPS Hotfix update. This hotfix signature update approach is quick and proactive which is necessary for containing zero-day attacks at right time. Moreover, the hotfix can be applied to other sensors without stopping or restarting their service. The Global Hotfix server uses Push technology to initiate the transaction. The client sensors have to subscribe to the Hotfix server for receiving updates. The Hotfix server provides live-update whenever a new signature is generated. It collects the new signature in a file and sends out to the client sensors. The signature file is sent over HTTP to client sensors. When a client sensor receives signature file, it calculate MD5 checksum. The result of the checksum is send to the Hotfix server. If the checksum doesn't match, the client discards the download and the server in response sends the same signature file again. The complete process is automatic that doesn't require and manual intervention. The best part of global update is that all the sensors remain updated and are in sync always. Figure 4 depicts this scenario where new signatures are pushed to various sensors.

4 Experimental Results

All experiments run on an isolated network in research lab. Honeynet comprises of Honeywall Roo-1.4 and high-interaction honeypots with Linux Sebek client installed on them. Tools like MetaSploit Framework, CLET and Admmutate are used to generate the payload for exploits. For IDS/IPS sensors SNORT is used. We have also developed a prototype for ZAD System with Signature Generator for our experiment. It is implemented in Java using Eclipse as an IDE and Mysql as a database. Four standard metrics were used to evaluate the performance of our technique: True Positive Rate (TPR), False Positive Rate (FPR), Total Accuracy (ACC) and Receiver Operating Characteristic (ROC) curve. TPR is the percentage of correctly identified malicious code shown in Equation 1. FPR is the percentage of wrongly identified benign code (Equation 1). ACC is the percentage of absolutely correctly identified code, either positive or negative, divided by the entire number of instances as shown in Equation 2. In ROC curve the TPR rate is plotted in function of the FPR for different points. In equations below, True Negative (TN) is the number of correctly identified benign code and False Negative (FN) is the number of wrongly identified malicious code.

$$TPR = \frac{|TP|}{|TP| + |FN|}; \quad FPR = \frac{|FP|}{|FP| + |TN|} \tag{1}$$

$$ACC = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|} \tag{2}$$

For obfuscated zero-day attacks we used zero-day polymorphic shellcodes. Some popular real-world polymorphic engines Metasploit, CLET and Admmutate were used to create polymorphic shellcodes that were unknown to our system. Total number of normal packets were 15453 out of which 734 were new shellcodes. Attack packets are only approximately 5% of the normal packets as zero-day attacks are rare events in reality. Table 1 shows the TPR, FPR, ACC and ROC area values for various polymorphic engines.

Table 1. System Detection Accuracy Against Polymorphic Engines

Polymorphic Engine	**TPR**	**FPR**	**ACC**	**ROC Area**
AdMmutate	0.84	0.034	0.84	0.87
Clet	0.92	0.027	0.91	0.95
Metasploit	0.91	0.031	0.91	0.93

Experiments were also conducted to measure response time of the system under different attack rates. We measured the response time as the time taken by the system to detect, analyze and generate signature for a new attack. Figure 5 depicts experimental results. It is found that with increase in attack rate the response time increases. This is because with increase in new attacks the time

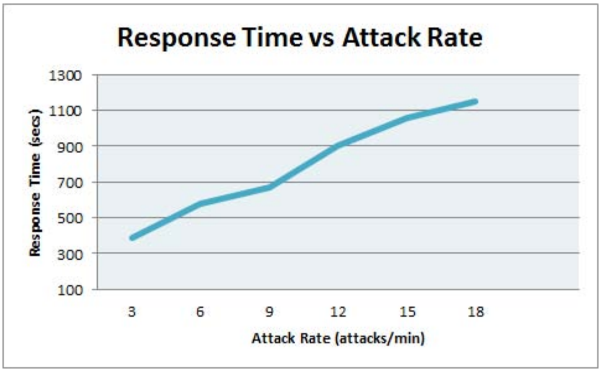


Fig. 5. Response Time vs Attack Rate

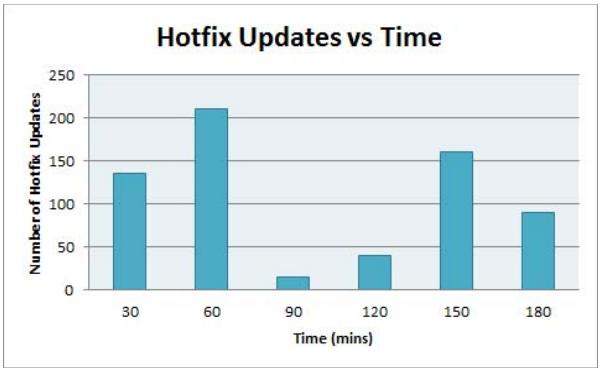


Fig. 6. Hotfix Updates vs Time

to handle these by the proposed system also increases. From our experiment we recorded a fairly good response time of one to two minutes for a single zero-day attack.

Separate experiment was conducted to check and verify signature updates. For this random number of attacks were launched every 30 mins against the system. In total 734 attacks were launched and 653 signature updates were seen. Figure 6 shows the number of hotfix signature updates with respect to time in an experiment. In first 60 mins number of hotfix updates are more as compared to update starting at 90th min. This update varies with respect to the number of attacks launched for that particular duration. In this experiment it is also found that the total number of attacks launched are not equal to the total number of hotfix signature updates. This difference is due to few false positives.

5 Conclusions

In this paper we have proposed a novel and efficient technique for detecting zero-day attacks. It clearly addresses the problems with previous approaches and provides an efficient solution to the whole problem. It provides: (i) An online detection mechanism against obfuscated zero-day attacks. (ii) Automatic evaluation of zero-day attacks at two levels thus, reducing false positive rate to near zero. (iii) Automatic signature generation with global hotfix update to various IDS/IPS sensors for containing new attack. Experimental results show detection rate of 89% with 0.03 false positive rate. The key problem of false positives is solved by using honeynet and by doing two-level attack validation. Besides the advantages, there are few limitations. Firstly, the system is checked for only one type of obfuscation i.e. polymorphism. Secondly, the signature generation process needs a better optimized algorithm. And thirdly to modify the technique to work for distributed and scalable systems. The future work includes addressing these limitations.

References

1. Bilge, L., Dumitras, T.: Before We Knew It: An Empirical Study of Zero-Day Attacks In The Real World. In: ACM Conference on Computer and Communications Security, pp. 833–844. ACM Press, New York (2012)
2. Symantec's Internet Threat Report of 2013 (2013), `https://scm.symantec.com/resources/istr18_en.pdf`
3. Crandall, J.R., Su, Z., Wu, S.F.: On Deriving Unknown Vulnerabilities from Zero-day Polymorphic and Metamorphic Worm Exploits. In: 12th ACM Conference on Computer and Communications Security, pp. 235–248. ACM Press, New York (2005)
4. Cheetancheri, S.: Collaborative Defense against Zero-day and Polymorphic Worms: Detection, Response and an Evaluation Framework. PhD Thesis, University of California (2007)
5. Schoelkopf, B., Platt, J., Shawe-Taylor, J., Smola, A., Williamson, R.: Estimating the Support of a High-Dimensional Distribution. J. Neural Computation 13(7), 1443–1471 (2001)
6. Sun, W.C., Chen, Y.M.: A Rough Set Approach for Automatic Key Attributes Identification of Zero-day Polymorphic Worms. J. Expert Systems with Applications 36(3), 4672–4679 (2009)
7. Almotairi, S., Clark, A., Mohay, G., Zimmermann, J.: A Technique for Detecting New Attacks in Low-Interaction Honeypot Traffic. In: 4th International Conference on Internet Monitoring and Protection, pp. 7–13. IEEE Computer Society, Washington, DC (2009)
8. Newsome, J., Karp, B., Song, D.: Polygraph: Automatically Generating Signatures for Polymorphic Worms. In: IEEE Symposium on Security and Privacy, pp. 226–241. IEEE Press, New York (2005)
9. Portokalidis, G., Bos, H.: SweetBait: Zero-hour Worm Detection and Containment using Low-and High-Interaction Honeypots. J. Computer and Telecommunications Networking 51(5), 1256–1274 (2007)

10. Wang, K., Cretu, G.F., Stolfo, S.J.: Anomalous Payload-based Worm Detection and Signature Generation. In: Valdes, A., Zamboni, D. (eds.) RAID 2005. LNCS, vol. 3858, pp. 227–246. Springer, Heidelberg (2006)
11. Kruegel, C., Kirda, E., Mutz, D., Robertson, W., Vigna, G.: Polymorphic Worm Detection using Structural Information of Executables. In: Valdes, A., Zamboni, D. (eds.) RAID 2005. LNCS, vol. 3858, pp. 207–226. Springer, Heidelberg (2006)
12. Wang, L., Li, Z., Chen, Y., Fu, Z., Li, X.: Thwarting Zero-day Polymorphic Worms with Network-level Length-based Signature Generation. J. IEEE/ACM Transactions on Networking (TON) 18(1), 53–66 (2010)
13. Polychronakis, M., Anagnostakis, K.G., Markatos, E.P.: Network-level Polymorphic Shellcode Detection using Emulation. J. Computer Virology 2(4), 257–274 (2006)
14. Leita, C., Dacier, M.: SGNET: A Distributed Infrastructure to Handle Zero-day Exploits. Research Report, EURECOM institute (2007)
15. Ting, C., Xiaosong, Z., Zhi, L.: A hybrid detection approach for zero-day polymorphic shellcodes. In: International Conference on E-Business and Information System Security, pp. 1–5. IEEE, Wuhan (2009)
16. Li, Z., Sanghi, M., Chen, Y., Kao, M.Y., Chavez, B.: Hamsa: Fast Signature Generation for Zero-day Polymorphic Worms with Provable Attack Resilience. In: Symposium on Security and Privacy, pp. 15–47. IEEE, Oakland (2006)
17. A 0-Day Attack Lasts On Average 10 Months, http://hackmageddon.com/2012/10/19/a-0-day-attack-lasts-on-average-10-months/
18. Polychronakis, M., Anagnostakis, K.G., Markatos, E.P.: Emulation-Based Detection of Non-self-contained Polymorphic Shellcode. In: Kruegel, C., Lippmann, R., Clark, A. (eds.) RAID 2007. LNCS, vol. 4637, pp. 87–106. Springer, Heidelberg (2007)
19. Alazab, M., Venkatraman, S., Watters, P., Alazab, M.: Zero-day Malware Detection based on Supervised Learning Algorithms of api call signatures. In: 9th Australasian Data Mining Conference (AusDM 2011), Ballarat, Australia, pp. 171–182 (2011)
20. Aleroud, A., Karabtis, G.: A Contextual Anomaly Detection Approach to Discover Zero-day Attacks. In: IEEE International Conference on Cyber Security (CYBERSECURITY 2012), Washington, pp. 40–15 (2012)
21. Jain, P., Sardana, A.: Defending Against Internet Worms Using Honeyfarm. In: CUBE International Information Technology Conference (CUBE 2012), Pune, India, pp. 795–800 (2012)
22. Tang, Y., Chen, S.: An Automated Signature-based Approach against Polymorphic Internet Worms. J. IEEE Transactions on Parallel and Distributed Systems 18(7), 879–892 (2007)
23. Comar, P.M., Liu, L., Saha, S., Tan, P.N., Nucci, A.: Combining Supervised and Unsupervised Learning for Zero-day Malware Detection. In: IEEE INFOCOM, Turin, pp. 2022–2030 (2013)
24. Aleroud, A., Karabtis, G.: Toward Zero-Day Attack Identification Using Linear Data Transformation Techniques. In: 7th IEEE International Conference on Software Security and Reliability (SERE 2013), Gaithersburg, MD, pp. 159–168 (2013)
25. Paul, S., Mishra, B.K.: PolyS: Network-based Signature Generation for Zero-day Polymorphic Worms. International Journal of Grid and Distributed Computing 6(4), 63–74 (2013)

A Theoretical Study on Access Control Model in Federated Systems

Anant V. Nimkar and Soumya K. Ghosh

Indian Institute of Technology,
Kharagpur, India 721 302
anantn@sit.iitkgp.ernet.in, skg@iitkgp.ac.in

Abstract. The federation is a special case of open system where the resources are controlled and accessed by cooperation of one or more roles in the federation. The federation system needs a few special treatments like a subset ownership (i.e. multiple user ownership) of the objects, dynamic access right allocation etc. The treatments can not be handled by any combination of mandatory, discretionary and role-based access control models (i.e. MAC, DAC and RBAC). This paper gives a theoretical study on an access control model in federating systems by analysing the nature of subjects, objects and their relationships; and then proposes a generic access control model for any federation system. The safety proof shows that the federation system always remains in a safe state using the proposed federation access control model.

Keywords: Access control models, Distributed security, Federation.

1 Introduction

The monolithic and proprietary technologies always give full-fledged and autonomous services but also return a few disadvantages which lead to customer-dissatisfaction e.g. a service-lock-in where the customers can not give up the provider and join another provider until the services agreement is over. The solution would be a federation where one or more entity transparently federates for their customers through brokered architecture. The federation gives a few advantages like a good quality of services, low cost services and all-time service availability etc. through a competitive business-market. There is very little literature related to the administration of federated resources, even though there are such advantages of the federation system.

The federation in the context of information technology is a special case of open system and works differently as against the traditional open system. The subjects, objects and the granular operations (i.e. access rights) are treated differently in the federation ecosystem as follows. First, a subject is not a singleton role but a subset of federating roles. Second, the federating entities collaboratively execute each granular operations after a successful federation is established among the federating entities. Third, the subject cannot take away objects in the federation because of *a subset ownership* in which the objects are owned

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 310–321, 2014.

by a subset of federating roles. Fourth, the federation ecosystem allocates and de-allocates the access rights to the subject over object when the federation is established and torn down respectively. The allocation and de-allocation of access rights are done dynamically and transparently.

The aforementioned special treatment of subject, object ownership and dynamic access right allocation in the federation fail the basic access control models (ACMs) as follows. In case of MAC, the designer first decides on the set of security levels and then they are always constant over the time after the decision. The resources are owned by the MAC's single hidden user (i.e. system). The resources are also administered by the system. In a nutshell, the MAC fails for the resource administration in the federation ecosystem. In case of standalone DAC, the resources can be taken away and owned by a single discretionary user. In case of the federation system, the objects cannot be taken away by the subject as it is a subset of federating roles. Hence the standalone DAC fails for the administration of federated resources. In case of RBAC, the single administrator decides on the allocation and de-allocation of access right on the object by the subjects. So it also fails to administrate the federated resources. The special treatments cannot be handled by any combinations of ACMs because the basic ACMs fail to administrate the federated resources.

In this paper, the guidelines for the formation of security labels of subject and objects are given by investigating the nature of subject, object ownership and their relationship in federation system. The proposed *federation access control model* uses the concept of composition of security labels of subject and object in the individual federating entities using a special operator. The model decides about object access by the subjects using two other special operators. These operators can be used to form any combination of security policies in terms of confidentiality and integrity. We will also give two examples of the federation system to show the generic nature of the proposed model.

The rest of the paper is organized as follows. The limitations of existing standard access control models for federation system and current related work is explored in Section 2. Section 3 elaborates the federation ecosystem in the context of *information technology*. The different treatments of subject, object and security labels in the federation ecosystem are given in Section 3.1, Section 3.2 and Section 3.3 respectively. The novel *federation access control* is presented in Section 4 along with primitive operations and the basics of information flow policies in Section 4.1 and Section 4.2 respectively. The two examples of the federation system, *IaaS cloud federation* and *electronic medical record federation* are presented in Section 5.1 and Section 5.2 respectively. The system safety proof and concluding remarks for the proposed federation access control model are given in Section 6 and Section 7 respectively.

2 Survey and Related Work

The confidentiality model [2] proposed by Bell and LaPadula does not vary security levels over the life-cycle of the system. The integrity model [3] proposed by Biba also does not vary security levels over the life-cycle of the system. The open/closed system using Chinese Wall security model [4] unknowingly uses a variable number of security levels — an ordered pair of ⟨*conflict-of-interest*, *company-dataset*⟩ — but does not provide the security levels/labels as a function of federating roles in the federation. The standard RBAC model [9] facilitates the variable number of security levels but is also delimited by 2^A, where A is the set of access rights. The RBAC can not allocate the security labels as a function of federating roles for open/closed system. In a nutshell, the standard access control models lack one or more special treatments of the federation.

The existing literatures on an access control model provide a few features for the federation ecosystem as follows. The mandatory access control model [10] merges the information flow policies for dynamism but it does not give the provision of a *subset ownership* of the objects to the subject required for the federation. Barker and Genovese propose a security framework [1] for the community/federation using the authorization of users and logical operators but it does not provide subset ownership over the objects. Wenchao et al. [12] extends RBAC access control model for the federation and proposes Role-Role mapping concept using the technique of Single-Sign-On (SSO). The confidentiality-aware federation access control model [6] decomposes the policies among tenants and providers for SaaS federation.

3 Federation Ecosystem

The federation of resources in the context of information technology can be defined as a federated service among a set of entities for the customers through a brokered architecture. An entity in the federation can be a service provider or service borrower. A service provider offers various services to its customers. A service borrower takes services of another service providers for its customers. Fig. 1 shows an abstract example of a federation between seven entities through a brokered architecture for the users. The circles, U_1 - U_8 are users and the irregular polygons, F_1 - F_7 are entities. The overlapping of the irregular polygons shows a federation among a set of entities for a particular user e.g. the federation $\{E_3, E_4, E_5\}$ for the user U_5 is shown by the overlapping of circle (i.e. U_5) and three irregular polygons (i.e. E_3, E_4 and E_5). The federation ecosystem inherently shows some potential security threats. First, the federation $\{E_3, E_4, E_5\}$ for the user U_5 may result in a direct information flow to the entities E_1, E_2 and E_6. Second, the federation $\{E_3, E_4, E_5\}$ for the user U_5 may result in an indirect information flow to all the entities.

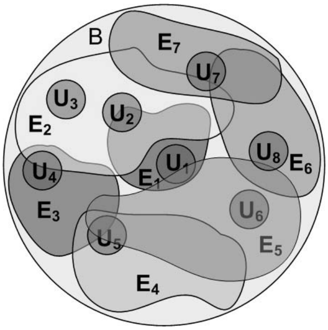


Fig. 1. An abstract federation among seven entities

3.1 Subject Namespace

Each federating entity autonomously maintains the namespace and security levels of customers/users and resources for the federation. The identity of a subject is shared with only federating entities after the successful federation. The identity of the subject is dynamically created and destroyed as the federation established and torn down respectively. The subject in federation is always a subset of $2^{entities \, \cup \, users}$. e.g. if there is a federation among E_3, E_4 and E_5 for the user U_5, then the subject namespace for the federation for the user U_5 is a subset of a power set of $\{E_3, E_4, E_5, U_5, B\}$ (i.e. $2^{\{E_3,E_4,E_5,U_5,B\}}$) where B is a broker for negotiating a federation for the users. We assume that the Single-Sign-On or distributed authorization [11] can be used to authorize such subjects. In a nutshell, the subjects in the federation are not the usual subjects of any open/closed system ACMs but they are the subset of $2^{entities \, \cup \, users}$.

3.2 Object Namespace

The federating entities keep a pool of resources which may be used for the federation among the entities. Each entity autonomously specifies the object's identity e.g. an entity E_3 may use r_1^3 as an identity of a resource. The federated resources are dynamically created and destroyed as the federation is established and torn down respectively. An identity of a federated resource is a subset of $2^{entities \, \cup \, users \, \cup \, resources}$ e.g. if there is a federation among E_3, E_4 and E_5 for the user U_5, then the object namespace for the federation of the user U_5 is a subset of $2^{\{E_3,E_4,E_5,U_5,R\}}$ where R is a set of potential resource pool for the federation.

3.3 Security Classes

The security labels of users and resources are assigned autonomously and statically by the entities. The security labels of subject and objects in the federation are formed using a special operator called cartesian union, $\uplus$. The number of types of roles in the federation creates different security classes. With this logic, the security labels in any federation can be formed by creating a n-tuple security label where n is the number of types of roles in the federation. If the abstract federation (Fig. 1) has four types of roles: i) service provider, ii) service borrower, iii) user, and iv) brokers then the security label will be $\langle SP, SB, U, B\rangle$ where SP, SB, U and B are security labels of service provider, service borrower, user and brokers respectively. Let us assume that E_3 is a service provider and; E_4 and E_5 are service borrowers. Let us also assume that the integer numbers are used as security labels in the present scenario. The subjects $\{E_3, E_4, E_5, U_5\}$ and $\{E_3, U_5\}$ would have $\langle\{3\}, \{4, 5\}, \{5\}\rangle$ and $\langle\{3\}, \{5\}, \emptyset\rangle$ security labels respectively.

4 Mathematical Model

In any federation ecosystem, there are two treatments for the management of security classes/security labels other than the special treatments mentioned earlier. First, the security classes in the federation access control model vary over the time. Second, the security labels/classes in the federation may not always form a lattice as per Denning information flow policy [7]. The reason for first treatment is due to the transparent federation required for the users. When the federation is formed between entities, the lattice among these federating entities is created and is removed when the federation is released. The reason for the second treatment will be given subsequently in the paper. All the aforementioned treatments results into a dynamic access control model for the federation system called as *Federation Access Control Model* (FACM) which must include all σ federating entities. So, the federation access control model can be formally defined as 5-tuple as follows.

$$F = \{(S_i, O_i, P_i, F_i, L_i) \mid 1 \leq i \leq \sigma\} \tag{1}$$

S_i is a set of subjects which is a subset of power set of all federating entities and users. O_i is a set of federated resources in the federation. Each subject can exercise a subset of 2^A access rights where A is a set of access rights. The potential execution of access rights can be viewed as 3-tuple set, $P_i \subseteq S_i \times O_i \times 2^A$. The set of security labels for the federated subjects and objects at particular time instance is a subset, F_i of $((S_i \to K) \times (O_i \to K))$ where K is a set of potential security labels in the federation. Each individual entity maintains a *partial order set* (*POSET*), $L_i = (B \subseteq K, \preceq)$ for the information flow policies in the federation. The set of security levels, B is dynamically managed by each entity in the federation system.

The FACM can use MAC and/or DAC rules to provide any basic security services in the form of confidentiality and/or integrity. The FACM uses two binary cartesian relations on the security labels to apply the MAC and/or DAC rules for the resource access decisions. The first binary relation $\subseteqq$ returns true if the first argument is a subset of the second argument and some other restrictions depending on a particular federation system. The binary relation $\equiv$ returns true if the two security labels are same and some other restriction depending on a particular federation system.

4.1 Primitive Operations

The FACM uses primitive operations to provide federation services by maintaining the data structure, *Access Control Matrix* in the federating entities. The execution of primitive operation results in the modification of some or all parts of the 5-tuple set. The federation services are executed synchronously in a subset of federating entities, $\{E_i \mid i \text{ is in federation for a particular user}\}$. The primitive operations operate on the set of subjects, objects and the *Access Control Matrix* which are represented by S, O and $a[s, o]$ respectively. The presented primitive operations use the capital alphabets (e.g. S, O etc.) and their primes (e.g. S', O' etc.) to show the elements of 5-tuple for a particular entity before and after the execution of primitive operations respectively. The synopses of all primitive operations are as follows.

(i) **create subject** s
Precondition: $s \notin S$ Postcondition: $S' = S \cup \{s\}, O' = O$
$(\forall y \in O')[a'[s,y] = \emptyset]$, $(\forall x \in S')(\forall y \in O')[a'[x,y] = a[x,y]]$
Synopsis: The subject s is created in ACMs of a subset of federating entities. This primitive operation will modify the subject set S_i, the security label function set F_i and the POSET L_i of a subset of federating entities. The security label of the subject s is the *cartesian union* of the individual entities' security labels in the federation.
(ii) **destroy subject** s
Precondition: $s \in S$ Postcondition: $S' = S - \{s\}, O' = O$
$(\forall y \in O')[a'[s,y] = \emptyset]$, $(\forall x \in S')(\forall \in O')[a'[x,y] = a[x,y]]$
Synopsis: The subject s is deleted from ACMs of a subset of federating entities. This primitive operation will modify the subject set S_i, the security label function set F_i and the POSET L_i of a subset of federating entities.
(iii) **create object** o
Precondition: $o \notin O$ Postcondition: $S' = S, O' = O \cup \{o\}$
$(\forall x \in S')[a'[x,o] = \emptyset$, $(\forall x \in S')(\forall y \in O')[a'[x,y] = a[x,y]]$
Synopsis: The object o is created in ACMs of a subset of federating entities. This primitive operation will modify the object set O_i, the security label function set F_i and the POSET set L_i of a subset of federating entities. The security label of the object o is the *cartesian union* of security labels of a subset of federating entities and the user.
(iv) **destroy object** o
Precondition: $o \in O$ Postcondition: $S' = S, O' = O - \{o\}$
$(\forall x \in S')[a'[x,o] = \emptyset$, $(\forall x \in S')(\forall y \in O')[a'[x,y] = a[x,y]]$
Synopsis: The object o is deleted from ACMs of a subset of federating entities. This primitive operation will modify the object set O_i, the security label function set F_i and the POSET set L_i of a subset of federating entities.
(v) **enter** $r \in A$ **into** $a[s,o]$
Precondition: $s \in S, o \in O$ Postcondition: $S' = S, O' = O$
$a'[s,o] = a[s,o] \cup \{r\}$, $(\forall x \in S')(\forall y \in O')[(x,y) \neq (s,o) \rightarrow a'[x,y] = a[x,y]]$
Synopsis: The attribute r is entered in ACMs of a subset of federating entities. This primitive operation will modify the security label function set F_i and the permission set P_i of a subset of federating entities.
(vi) **delete** $r \in A$ **from** $a[s,o]$
Precondition: $s \in S, o \in O$ Postcondition: $S' = S, O' = O$
$a'[s,o] = a[s,o] - \{r\}$, $(\forall x \in S')(\forall y \in O')[(x,y) \neq (s,o) \rightarrow a'[x,y] = a[x,y]]$
Synopsis: The attribute r is deleted from the ACM of a subset of federating entities. This primitive operation will modify the security label function set F_i and the permission set P_i of a subset of federating entities.

4.2 Information Flow Policies

The FACM uses two cartesian operators, cartesian subset ($\subseteqq$) and cartesian equal ($\equiv$); and a set of access rights. The information flow polices for the particular federation system can be designed as per the requirements by defining the meaning of the two cartesian operators and access rights. This model can give any combination of basic security services of confidentiality and/or integrity by applying the meaning to the granular operations of the particular federation system. We will later give two examples by attaching the meaning to the operators and the access rights to give the reason for generic nature of the FACM.

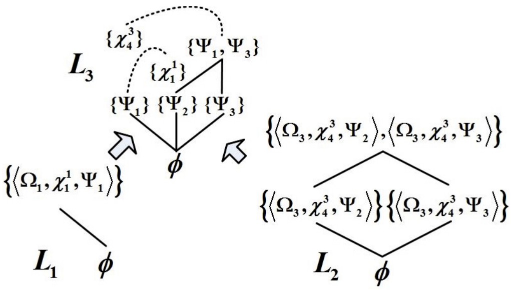

$\{\chi_4^3\}$
$\{\Psi_1,\Psi_3\}$
L_3
$\{\chi_1^1\}$
$\{\Psi_1\}$ $\{\Psi_2\}$ $\{\Psi_3\}$
$\{\langle\Omega_3,\chi_4^3,\Psi_2\rangle,\langle\Omega_3,\chi_4^3,\Psi_3\rangle\}$
ϕ
$\{\langle\Omega_1,\chi_1^1,\Psi_1\rangle\}$
$\{\langle\Omega_3,\chi_4^3,\Psi_2\rangle\}$ $\{\langle\Omega_3,\chi_4^3,\Psi_3\rangle\}$
L_1 ϕ
L_2 ϕ

Fig. 2. The lattices for users and corresponding POSET

The security levels always form a lattice in access control models for any open/closed system as per Denning's information flow policy [7]. The same does not hold in the federation system, if we view a federation among the entities as a global point of view. But the security levels for a particular user always form a lattice in the federating entities. The security label in the abstract federation has a form of $\langle \Omega_x, \chi_x^y, \Psi_z \rangle$ where χ_x^y, Ω_x and Ψ_z are security labels of a service borrower, a user and a set of service providers for the federation. Fig 2 shows lattices L_1 and L_2 for the users U_1^1 and U_3^4 respectively. The compressed *POSET* L_3 of all the established federations (i.e. L_1, L_2) in a particular entity gives a few advantages e.g. it takes less space and also shows any potential information flow between different users.

5 Federation System Examples

In this section, we give two examples of the federation system from different domains to give the reasons for the generic nature of the proposed model. The first example – *IaaS cloud federation* – is from the domain of cloud computing where the system requires confidentiality among the users and integrity of software. The second example – *Electronic Medical Record Federation* – is from the domain of information technology where the system requires that the patients' medical information must be confidential as per patients' wishes.

5.1 IaaS Cloud Federation

An IaaS cloud federation is a federated cloud service of virtual infrastructures from a set of InPs and SePs with full server and network virtualization [5]. There are four federating roles namely SePs, InPs, Users and the broker; and two types of federated resources: virtual nodes and virtual links. The service borrower and provider are called SeP and InP respectively. The 3-tuple security label has the form of $\langle SeP_{SL}, U_{SL}, InP_{SL} \rangle$ where SeP_{SL}, U_{SL} and InP_{SL} are the security labels of SeP, user and a set of InPs respectively.

Confidentiality and Integrity Policy. The IaaS cloud federation requires *read* ($\underline{r}$), *write* ($\underline{w}$), *execute* ($\underline{e}$) and *federation* ($\underline{f}$) access rights. The meaning of *read* and *write* stands for receiving and sending the packets respectively on/to the link/node. The *execute* access right has a usual meaning to execute a federation service to configure a node/link. The *federation* ($\underline{f}$) is special access right to the special subject on a special type of object to show a successful establishment of the federation.

The IaaS cloud federation requires confidentiality among the users and entities; and software integrity within federating entities and users. The second basic security service, *integrity* of service/software in the federation is treated differently. The integrity level is higher as the number of federating entities is more in the subject. The binary relation for the IaaS cloud federation are extended as follows. The binary relation $\subseteqq$ returns true if the first two fields of the security labels are non-empty and same, and third field of the first security label is a subset of the third field of the second security label. The binary relation $\equiv$ returns true if the first two fields of the security labels are non-empty and same, and the third field of first security label is same as the third field of the second security label. The MAC and DAC rules for controlling access to federated objects of IaaS cloud federation are given as follows.

(i) The subject $s \in S_i$ can *read* object $o \in O_j$ for $i = j$ or $i \neq j$ if and only if, a) $\forall(f \in S)\exists f(s \subseteqq f)$ and $\underline{f} \in a[f, f]$ b) $SL(s) \subseteqq SL(f)$ c) $SL(o) \subseteqq SL(f)$ and d) $\underline{r} \in a[s, o]$

(ii) The subject $s \in S_i$ can *write* object $o \in O_j$ for $i = j$ or $i \neq j$ if and only if, a) $\forall(f \in S)\exists f(s \subseteqq f)$ and $\underline{f} \in a[f, f]$ b) $SL(s) \subseteqq SL(f)$ c) $SL(o) \subseteqq SL(f)$ and d) $\underline{w} \in a[s, o]$

(iii) The subject $s_1 \in S_i$ can *execute* object $s_2 \in S_j$ for $i = j$ or $i \neq j$ if and only if, a) $\forall(f_1 \in S)\forall(f_2 \in S)\exists f_1(s_1 \subseteq f_1)\exists f_2(s_2 \subseteq f_2)$ and $(SL(f_1) \equiv SL(f_2))$ and $\underline{f} \in a[f_1, f_1]$ and $\underline{f} \in a[f_2, f_2]$ b) $SL(s_1) \subseteqq SL(s_2)$ and c) $\underline{e} \in a[s_1, s_2]$

The first two rules allow any information flow between only federating InPs and SeP; and confidentiality between non-federating InPs and SeP. The third rule gives software integrity in terms of trust between the subjects of federating InPs and SeP.

Federation Services. The IaaS cloud federation needs services for various cooperative operations like create/delete federation, create/delete virtual nodes and create/delete virtual links etc. These services are executed by the federation system after the subject authorization.

5.2 Electronic Medical Record Federation

The patients' health information can be electronically stored at different locations. They are owned and managed by multiple stakeholders and/or the patients. The Electronic Medical Record (EMR) federation [8,13] is a federated service of patients' medical information from a set of *care delivery organization* (CDO), medical practitioners (MP) and patients (P). The EMR federation has three types of medical records : electronic medical record (EMR), electronic health record (EHR), Personal Health Record (PHR). The MPs and CDOs maintain patients medical information in the EMR. The patients maintain PHR and EHR which is a subset of EMR records. The 3-tuple security label has the form of $\langle P_{SL}, CDO_{SL}, MP_{SL} \rangle$ where P_{SL}, CDO_{SL} and MP_{SL} are the security labels of patients, care delivery organization and medical practitioner respectively.

Confidentiality Policy. The EMR federation requires *read* ($\underline{r}$), *write* ($\underline{w}$) and *federation* ($\underline{f}$) access rights. The meaning of *read* and *write* are same in the usual context of information technology. The EMR federation requires confidentiality of medical records of the patients. The EMR federation requires that the creator of the record should be able to write and the federated roles can only read the medical records. The binary relation of FACM for the EMR federation are extended as follows. The binary relation $\equiv$ returns true if there is at least one field of arguments of security labels are non-empty and same. The binary relation $\subseteqq$ returns true if there is at least one field of both arguments is non-empty and; the first argument is a cartesian subset of the second argument. The MAC and DAC rules for controlling access to federated objects in EMR federation are given as follows.

(i) The subject $s \in S_i$ can *read* object $o \in O_j$ for $i = j$ or $i \neq j$ if and only if, a) $\forall(f \in S)\exists f(s \subseteqq f)$ and $\underline{f} \in a[f, f]$ b) $SL(s) \subseteqq SL(f)$ c) $SL(o) \subseteqq SL(f)$ and d) $\underline{r} \in a[s, o]$ e) $SL(s) \subseteqq SL(o)$

(ii) The subject $s \in S_i$ can *write* object $o \in O_j$ for $i = j$ or $i \neq j$ if and only if, a) $\forall(f \in S)\exists f(s \subseteqq f)$ and $\underline{f} \in a[f, f]$ b) $SL(s) \subseteqq SL(f)$ c) $SL(o) \subseteqq SL(f)$ and d) $\underline{w} \in a[s, o]$ e) $SL(s) \equiv SL(o)$

The first rule allow any information flow only in one direction because the second rule does not allow the flow of information in the opposite direction and hence give the confidentiality of medical records as per their creators.

Federation Services. The EMR federation system needs various cooperative operations like crate/delete federation, create/update/delete/view the medical records (i.e. PHR, EMR and EHR etc.) It also uses few administrative authorization services to authorize the subjects.

6 System Safety

Any federation system can be modelled as a function $W : R \times V \rightarrow D \times V$ where the R is a set of input requests, D is a set of outputs and V is a set of states. The state of the federation system, $v \in V$ includes the internal variables $\{(S_i, O_i, P_i, F_i, L_i) \mid 1 \leq i \leq \sigma\}$ at a particular instance. The main problem with this kind of modelling is that the number of states is not fixed as shown in Fig. 3, so the simple *Finite State Machine* can not be used. But the federation system can be modelled, if the input requests, output decision and the system states are recorded. Fig. 4 shows the modelling of the system through a function $\Sigma : X \times Z \rightarrow Y \times Z$ where the sets X, Y and Z are sets of sequences of input requests, output decisions and states respectively over the time. The transition function of the federation system is $\delta : R \times W \cup z_0 \rightarrow D \times W$ where z_0 is a special initial state of the system where all elements of 5-tuple model are equal to $\emptyset$.

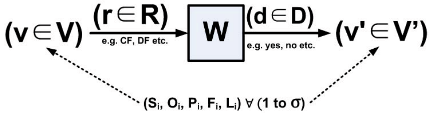


Fig. 3. A federation system modelling as $W : R \times V \rightarrow D \times V$

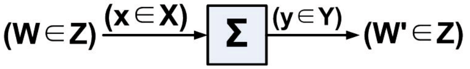


Fig. 4. A federation system modelling as $\Sigma : X \times Z \rightarrow Y \times Z$

The federation system is said to be un-secure if any state $v \in V$ does not satisfy the MAC and/or DAC rules specified in Section 5.1/5.2. The proposed FACM can be proved in two steps. In the first step, the system must be in a safe state after the execution of defined security policies for the particular federation system. In the second step, the execution of a sequence of services must stay the system in a safe state. The first step depends on the policies of a particular federation system, so we prove the first step for both examples *IaaS cloud federation* and *Electronic Medical Record Federation* in Theorem 2 and Theorem 1 respectively. The second step is proved as Theorem 3 which is independent of any particular federation system.

Theorem 1. *The MAC and DAC rules for the IaaS cloud federation system, F_{IAASF} as given in Section 5.1 provides confidentially among users and integrity for the federation services.*

Proof. The implication of the statements can be proved using the information flow lattices for the users. If there exist a user U_3^2 which can read and write the objects, then there must exist the subjects $\{S_2, U_3^2, *_1\}$ and its information flow lattice. There must also be a federation for the user of the form $\langle \Omega_2, \chi_3^2, *_2 \rangle$ such that $*_1 \subseteq *_2$. This proves the clauses (a),(b) and (c) of statements (i) and (ii) and the clause (a) of statement (iii). If the subject can read and write then there must exist $\underline{r}$ and $\underline{w}$ access rights for the subject $\{P_1, *_1, *_2\}$ which prove the existence of the clauses clauses (d) of statements (i) and (ii). Finally the federation system provides confidentiality, so the information must be allowed to flow in the security class $\langle \Omega_2, \chi_3^2, *_3 \rangle$ where $*_3$ is a subset of power set of $*_2$. If the subject, s_1 can execute the subject s_2 then the clauses (b)-(c) of statement (iii) must be true.

The converse of the statements can be proved as follows. The first statement (i.e. (i)) allow the flow of information in one direction using fifth DAC clause (i.e. (e)) and existence of federation because of remaining clauses (i.e. (a)-(d)). The second statement (i.e. (ii)) blocks the flow of information in opposite direction using fifth DAC clause (i.e. (e)) and the existence of federation because of remaining MAC clauses (i.e. (a)-(d)). The third statement (i.e. (iii)) allow the flow of information only from high to low integrity level using fifth DAC clause (i.e. (e)) by executing more trusted federation services and the existence of federation because of remaining MAC clauses (i.e. (a)-(d)).

Theorem 2. *The MAC and DAC rules for the EMR federation system, F_{EMR} as given in Section 5.2 provides confidentially among the federated patients.*

Proof. The implication of the statements can be proved using the information flow lattices for the patients. If there exist a patient P_1 which can read and write the objects, then there must exist the subjects $\{P_1, *_1, *_2\}$ and it's information flow lattice. There must also be a federation for the patients in the form $\{P_1, *_3, *_4\}$ such that $*_1 \subseteq *_4$ and $*_2 \subseteq *_4$ (i.e. clauses (a),(b) and (c) of statements (i) and (ii)). If the subject can read and write then there must exist $\underline{r}$ and $\underline{w}$ access rights for the subject $\{P_1, *_1, *_2\}$ which prove the existence of the clauses clauses (d) of statements (i) and (ii). Finally the federation system provides confidentiality, so there must exist $SL(\{P_1, *_1, *_2\}) \equiv SL(o_1)$ and $SL(\{P_1, *_1, *_2\}) \subseteqq SL(o_2)$, which prove the clause (e) of statements (i) and (ii) for some objects o_1 and o_2 which satisfies these conditions.

The converse of the statements can be proved as follows. The first statement (i.e. (i)) allow the flow of information in one direction using fifth DAC clause (i.e. (e)) and existence of federation because of remaining clauses (i.e. (a)-(d)). The second statement (i.e. (ii)) blocks the flow of information in opposite direction using fifth DAC clause (i.e. (e)) and the existence of federation because of remaining MAC clauses (i.e. (a)-(d)).

Theorem 3. *FACM maintains the federation system — modelled using Σ , W and δ functions — in a safe state over all the time.*

Proof. Initially, all the 5-tuple elements are empty hence the initial state, z_0 is empty. Let us assume that the system goes safely from $v \in V^{n-1}$ to $v' \in V^n$ for any large integer number n using the MAC and DAC rules of F_{EMR}/F_{IAASF} federation system. The theorem can be proved by induction and contradiction. Let us assume the system goes to an unsafe state after k^{th} step. If the system goes to unsafe state from $(k-1)^{th}$ to k^{th} for any $k < n$ and $k > 1$, then it contradicts Theorem 1/2. So all transition must be safe and hence the FACM maintains the system in safe state over all the time.

7 Conclusion

The resource access decision in federation system is done collectively by a subset of federating roles. This results in a few special treatments to the subjects and objects related to access control model. We investigated the various treatments on the subject, object ownership and their relationship in the federation and gave the guidelines for the formation of security labels of subjects and objects for the federation access control model. We also proposed a novel security model for federation system which provides any combination of basic security services like confidentiality and/or integrity. The system safety proof shows that the FACM is secured over all the time.

References

1. Barker, S., Genovese, V.: Secommunity: A framework for distributed access control. In: Delgrande, J.P., Faber, W. (eds.) LPNMR 2011. LNCS, vol. 6645, pp. 297–303. Springer, Heidelberg (2011)
2. Bell, D., LaPadula, L.: Secure computer systems: Mathematical foundations. Technical Report MTR-2547, vol. I. MITRE Corporation, Bedford (1973)
3. Biba, J.K.: Integrity considerations for secure computer systems. MITRE Co., technical report ESD-TR 76-372, pp. 1–68 (April 1977)
4. Brewer, D.F.C., Nash, M.J.: The chinese wall security policy. In: IEEE Symposium on Security and Privacy, pp. 206–214 (1989)
5. Buyya, R., Ranjan, R., Calheiros, R.N.: Intercloud: utility-oriented federation of cloud computing environments for scaling of application services. In: Hsu, C.-H., Yang, L.T., Park, J.H., Yeo, S.-S. (eds.) ICA3PP 2010, Part I. LNCS, vol. 6081, pp. 13–31. Springer, Heidelberg (2010)
6. Decat, M., Lagaisse, B., Joosen, W.: Toward efficient and confidentiality-aware federation of access control policies. In: Proceedings of the 7th Workshop on Middleware for Next Generation Internet Computing, MW4NG 2012, pp. 4:1–4:6. ACM, NY (2012)
7. Denning, D.E.: A lattice model of secure information flow. Commun. ACM 19(5), 236–243 (1976)
8. Dong, G., Cui, G., Shi, W., Miao, Y.: Community health records and hospital medical record file sharing system model. In: 2011 IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS), pp. 146–148 (2011)
9. Ferraiolo, D., Kuhn, D.: Role-based access control. In: 15th National Computer Security Conference, pp. 554–563 (October 1992)
10. Rao, V., Jaeger, T.: Dynamic mandatory access control for multiple stakeholders. In: Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, SACMAT 2009, pp. 53–62. ACM, New York (2009)
11. Wang, S., Zhang, Y.: A formalization of distributed authorization with delegation. In: Boyd, C., González Nieto, J.M. (eds.) ACISP 2005. LNCS, vol. 3574, pp. 303–315. Springer, Heidelberg (2005)
12. Wenchao, Z., Yafen, L.: Federation access control model based on web-service. In: 2010 International Conference on E-Business and E-Government (ICEE), pp. 38–41 (2010)
13. Zhang, R., Liu, L.: Security models and requirements for healthcare application clouds. In: 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 268–275 (2010)

CASSHH – Case Adaptive SSH Honeypot

Adrian Pauna and Victor Valeriu Patriciu

Military Technical Academy, Bd. George Cosbuc 81-83,
Bucharest, Romania
adrian.pauna.ro@gmail.com, vip@mta.ro

Abstract. This paper presents a novel honeypot system implemented as a deliberative agent, built by means of a Case Based Reasoning (CBR) system. The honeypot system uses as reference an existing medium interaction honeypot (Kippo) and leverages the Beliefs-Desires-Intentions (BDI) deliberative agents improved with the learning capabilities of Case Base Reasoning (CBR) technique. The main goal is to create an autonomous system capable to learn and adapt by interaction with the attackers. The preliminary experimental results show that the developed system reacts as it was foreseen.

Keywords: honeypot systems, Belief-Desire-Intention, Intelligent Agents, Case Based Reasoning.

1 Introduction

In the recent years, the use of honeypot systems has grown dramatically. One of the honeypot systems that became a sort of a star is the medium interaction honeypot system called Kippo [1].

Almost all the organizations involved in the collection of information about attackers and malware use Kippo beside other well-known honeypot systems (Dionaea, Honeyd, etc.).

Recently Jose Nazario [2], a recognized expert in the area of honeypot systems, released a short study on the captured logs of a personal Kippo instance. The main question he tries to give an answer in his study is “How can I get users to engage with the honeypot more?” Beside this question the author also establishes as a target for research: the improvement of the existing honeypot system so to gather more data about attack and attackers.

In this paper we present a new developed honeypot system called CASSHH (Case Adaptive SSH Honeypot) that uses as basis the source code of the medium interaction honeypot system Kippo and implements adaptation capabilities by means of Case Based Reasoning – Belief Desire Intention Agents (CBR -BDI). In section 2 we present a short summary of the well-known honeypot system Kippo. In Section 3 we debrief you about the Case Based Reasoning elements and in the following section (Section 4) we give an insight on the Belief-Desire-Intention agents. The 5th Section is very important for our research and we detail the merge between CBR and BDI agents that we used for the development of our honeypot system. CBR BDI is a new approach for the implementation of BDI agents in which they are mapped into CBR

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 322–333, 2014.

systems. Section 6 presents the relevant aspects regarding the implementation of the CASSHH system talking about the creation of the action capabilities, CBR-BDI module and the modeling of the lifecycle of the resulting BDI agent. In section 7 we present a summary of the results we have obtained during the tests we have done and in section 8 we conclude and present the future improvements there should be done on the presented honeypot system.

2 SSH Honeypot Systems

The SSH protocol is presented in RFC 4251 as "a protocol for secure remote login and other secure network services over an insecure network"[3]. Therefore it is widely used by system administrators as a replacement for the old and insecure Telnet or other “r” commands such as rsh and rcp.

Used for remote access and remote file transfer SSH protocol is prone to brute force attacks with the scope of compromising the system accounts used for login. In this context the creation of a SSH Honeypot system was natural. The system that managed to be a real success was Kippo.

Developed by Upi Tamminen [4], the honeypot system offers several interesting capabilities such a fake filesystem, logging of all the commands the attacker enters, implementation of a number of commands that allow gathering of malware code downloaded by the attackers.

The system was created in Python and uses well known libraries for networking such as Twisted [5].From the perspective of the level of interaction of honeypot systems it is considered a medium one, because it does not provide a complete access to the underlying operating system, as high interaction honeypots do. But also it does not offer access just to the SSH service, such as low interaction honeypots do, instead it provides an emulated SSH terminal very similar to the one a normal SSH server has.

3 Case Based Reasoning (CBR)

As Leake et al [6] stated “Case-based reasoning is [...] reasoning by remembering”. A Case Based Reasoning system solves problems by using solutions used for similar previous ones. In principle, CBR acts similar to human reasoning when solving problems based on previous cases. The main logic of CBR system resides in the four phase the process has:

- **Retrieve** from a database the most similar case (or cases);
- **Reuse** the case selected in the idea of trying to solve the current problem;
- **Revise** and adapt the decided solution if this is necessary
- **Retain** the final solution as a new case in the database.

Different methods of organizing, retrieving, utilizing and indexing the knowledge contained by the past cases have been developed in the recent years.

The usage of CBR in the area of Information security has been related with the study of Intrusion detection [7] but also in the honeypot systems area [8].

4 Belief-Desire-Intention (BDI) Agents

The Belief-Desire-Intention (BDI) agent model provides reactive and proactive behavior using an event-driven execution model. At the basis of BDI agent stays the simplified view of human intelligence. Formally, the agent has a view of the world (Beliefs), a number of goals it wants to achieve (Desires) and it establishes Plans (Intentions) to act on this using the accumulated experience. In the recent years, a big attention was given to address the specification, design, verification and applications of Belief-Desire-intention agents (BDI). Based on their internal mental state, BDI agents take actions to affect their environment. The whole process is based on the continuous received input and relies on the fact that agents are situated in a changing environment. The primary mental attitudes of the agents are captured at informational level through beliefs, at motivational level through desires and at decisional level through intentions.

BDI agents have been used in several implementations of critical applications [9, 10, 11 and 12].

5 BDI-CBR Agents

An intelligent agent (IA) is considered an autonomous entity capable to observe through sensors and to act upon an environment using actuators. If the agent directs its activity in the scope of achieving specific goals it is considered rational. In order to achieve their goals intelligent agents may also learn or use knowledge.

Based on their characteristics agents can be classified in two different ways [13]:

1. From the mobility perspective:

- Mobile agents – if they are able to migrate to different nodes in a network
- Static agents – if they are not able to migrate

2. From their behavior perspective :

- Reactive and deliberative if they have a deterministic model
- Based on a reasoning system – if for example they are built based on a rule based system.

BDI-CBR agents, also called deliberative agents, are used to implement adaptive systems.

Laza et al [14] showed how deliberative agents can be built by means of a case-based reasoning system. The proposed system functioning is guided by a human expert and can be used for providing advice. As stated in the paper "Agents must be able to reply to events, which take place in their environment, to take the initiative according to their goals, to interact with other agents (even human), and to use past experiences to achieve present goals".

In the case of deliberative agents the BDI model formalizes as it follows:

- The belief is what the agent knows about himself and about the its environment;
- The desire is what the agent tries to achieve;
- The intention is a sequence of actions and can be identified as plans.

The authors present also the main, two problems a deliberative agent faces:

- The difficulty of finding a mechanism to permit an efficient implementation;
- The lack of learning capability , which is a basic requirement since a BDI model needs to constantly add, modify, eliminate beliefs, desires and intentions.

As stated in the paper the solution for these problems is the implementation of a BDI agent using a case-based reasoning model. This model also facilitates learning and adaptation.

The main idea is to have a direct mapping between the agent conceptualization of the system and its implementation in the form of CBR system that automatically provides a plan adjusted subsequently by a human expert.

As depicted in Figure 1 the structure of a CBR system is modeled around the concept of case mapped on the structure of BDI agent. The case base that stores the cases of past believes, desires and intentions of the CBR system are the agent's knowledge-base.

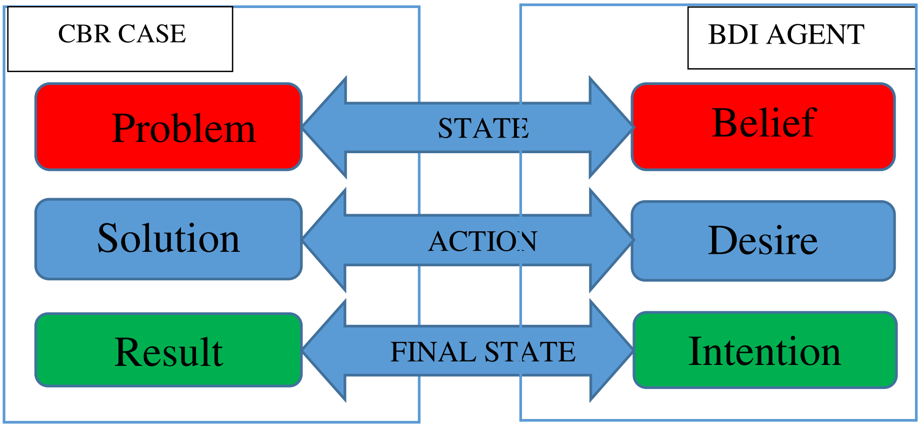


Fig. 1. CBR system mapping on BDI agent

6 Design and Implementation

In this section, we discuss the design and the implementation of CASSHH. We present our intended goals and how they were implemented.

From the recently studies on the profiling of attackers behavior after they compromise a SSH server, it is obvious that attackers download their own software tools that they use to issue further attacks [15].

We expect our honeypot system to be able to adapt to the commands issued by an attackers that compromises the emulated SSH service, so to deceive her to download as much as possible software.

CASSHH must be able to interact with the attackers by the means of the five actions implemented in the system:

- Allowing execution of a command
- Blocking the execution of a command
- Delaying the execution of a command
- Faking the output of a command
- Insulting the attackers.

The following code presents the implementation of the actions module of the honeypot system:

```
CURRENT_CASE["initial_cmd"] = self.honeypot.cmd

  action = self.getAction()
  CURRENT_CASE["action"] = action
  # Allow
  if action == 0:
          self.call()
      # Delay
      elif action == 1:
          self.write("delay ...\n")
          time.sleep(3)
          self.call()
      # Fake
      elif action == 2:
          ret =
getCasshDB().getFakeCommand(self.honeypot.cmd)
          if len(ret) and ret[-1] != "\n":
        ret = ret + "\n"
          self.write(ret)
      # Insult
      elif action == 3:
          location = ipLocation(self.honeypot.clientIP)
          self.write("Insult Message! IP=
%s/location=%s\n" % (self.honeypot.clientIP, location) )
          ret = getCasshDB().getInsultMsg(location)
          self.write(ret)
      elif action == 4:
          self.write("Blocked command!\n")

        self.exit()

    def isAllow(self):
```

CASSHH is developed in python and has as a reference the existing SSH honeypot Kippo. CASSHH uses the source code of Kippo on top of which we have implemented the actions module and the BDI-CBR module.

6.1 Architecture of the Honeypot System

As we can see in Figure 2, the system integrates different technologies, so to provide the necessary merge between honeypot system capabilities and intelligent agents' technology.

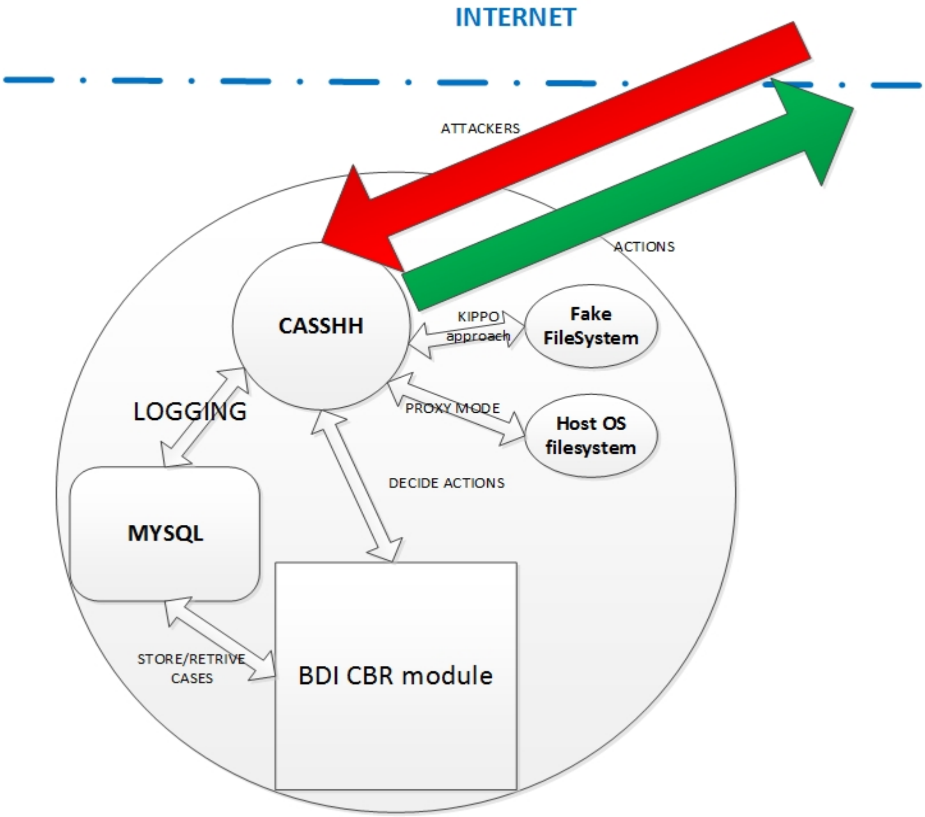


Fig. 2. CASSHH functional scheme

The BDI-CBR agent from the perspective of a Honeypot system has the following features:

- Emulated SSH service – uses the Kippo implementation based on twisted python library and emulates a SSH server that offers access to the underlying Linux operating system in two modes:

  - ○ Proxy-mode : CASSHH offers access to the real commands(ex. ls, pwd,uname)
  - ○ Emulated-mode : CASSH offers access to code implemented commands (ping, ifconfig, ssh, sftp, wget)

- Access to a limited number of commands – the emulated server offers access to a limited number of 75 Linux commands. They were selected according to the level of usage by hackers. Basically we have selected the most common commands corresponding to the profiles as listed in the table form the Annex.
- A logging module – the emulated server logs all the SSH clients types, authentications and inputs in the terminal
- An actions module – the server blocks, delays, fakes the output of the commands and insults the attackers based on their IP Geolocation.

6.2 Architecture of the Agent System

CASSHH implements the approach of a deliberative agent through the use of a BDI-CBR model. A CBR system follows the specific four “R” phases: Retrieve Reuse, Revise and Retain as you can see in figure 3.

The CBR-BDI agent is described by the 7-tuple <E, GAL, CM, PAL, EK, O, M>. Because of the direct correspondence the notation can be used to define beliefs, desires and intentions [14].

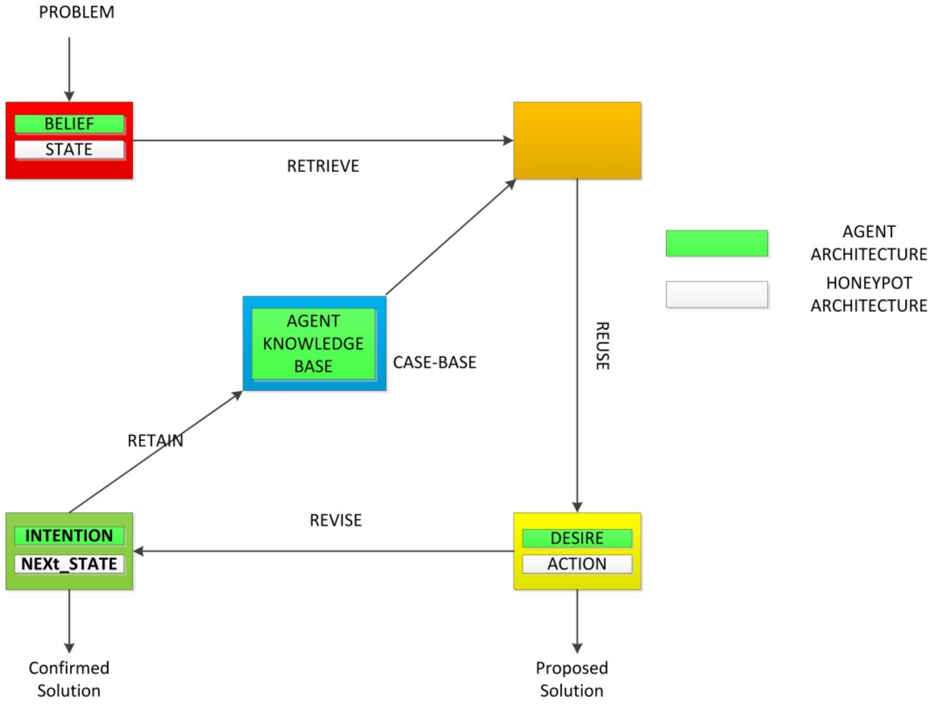


Fig. 3. Honeypot Agent Architecture

For our research we have used the following description of the notations is used proposed by Leza et al [14]:

- E: describes the current environment and is denoted as E = <e_1, e_2, ...,e_n> where e_i ∈ E, i ∈ {1,2,...,75}. Variables e_i are defined by a tuple <*$name\_var_i$*, *$value_i$* > where :
    - o *$name\_var_i$* is the name of the variable (case attribute) and the attributes are :

initial_command, profile_of_command, action, next_command

-
    - o *$value_i$* is its correspondent value, according to the description of each attributes we have Linux commands, commands profiles according to existing behavior profiling studies, the implemented actions;
- GAL : General actions Library is described by GAL = <ga_1, ga_2, ..., ga_p> where ga_i ∈ GAL, i ∈ {1,2,...,5}
- CM: Case Memory stores sets of previous cases CM = <c_1, c_2, ...,c_k>, where each case c_i is formulated as a 3-tuple <*B*, *D*, *I*> representing past beliefs, past desires and past intentions.
- **PAL**: Producer Actions Library is a collection of actions. Each intention is an ordered sequence, *I* = <a_1, a_2, ..., a_o> where each change from state to state is produced after carrying out an action, a_i ∈ PAL, i ∈ {1,2,...,5}, a_i is a tuple <*$Name_i$*, $Next\_command_i$> where *$Name_i$* ∈ *ga_i*, *ga_i* is a general action from general actions library GAL, *$Next\_command_i$* is a set commands executed after *ga_i*.
- **EK**: Expert knowledge EK is composed of a set of default rules associated with case adaptation and case retention process. For our implementation the adaption is done manually by a human expert.
- **O**: A set of current goals (O) in a particular belief-world. The objectives are appropriate final states in the environment. O = <o_1, o_2, ..., o_t>, where o_i ∈ *B* and *i* ∈ {1,2,...,75}, *O* ⊂ *B*. This set is null if the environment is not definite.
- **M**: Set of functions of similitude. A similarity function determines the degree of equality between two states. M = <m_1, m_2, ..., m_j>, where m_i ∈ *M* and *i* ∈ {1,2,...,j}.

6.3 Implementation of the System

For our implementation:

- E, the environment is composed from the 75 set of Linux commands available to the attacker.
- GAL, the general actions library is composed of the five actions presented and implemented in the Honeypot application.
- CM: Case Memory is implemented using a Mysql database in which we store the cases under the format:
- **O**: the set of current goals is related to the goal of our honeypot system regarding the deceiving of attackers to download as much as possible software used for further attacks.
- **M**: the set of functions of similitude is composed of the similarity functions used in the Jcolibri framework [16].

7 Results

The honeypot system described above was tested on an EC2 machine will full Internet access during three months of 2013 [17]. The system is not fully operational and because the aim of the project was to develop a research prototype and not a commercial tool. From the technical and scientific point of view the initial results have been very successful. During the first month the honeypot system collected four pieces of software downloaded by the attackers. On the other hand as we expect the system was able to interact autonomously with the attackers and learned which commands should not be blocked since they permit attackers to get software from different locations (via ftp,http,ssh).

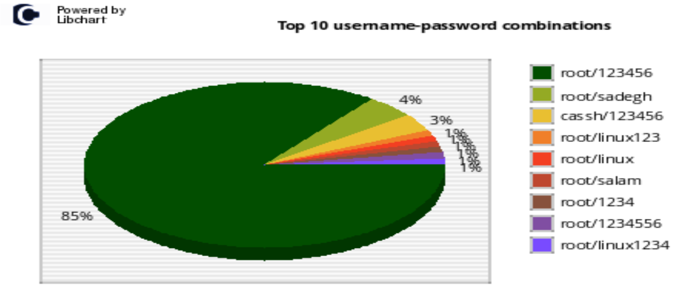


Fig. 4. Brute force attacks on our honeypot system

As expected our honeypot was targeted and brute forced. After access was gained attackers issued commands and some of them were deceived to download software.

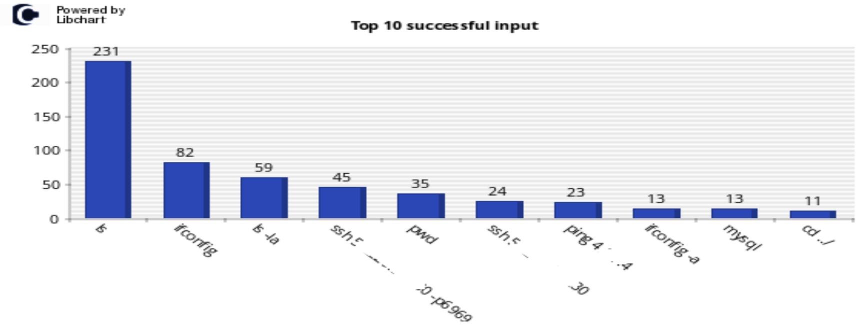


Fig. 5. Commands executed by the attackers

The formalism defined in [14] facilitates the straight mapping between the agent definition and the CBR construction. The concept of case is fundamental when working with CBR, therefore the definition of a case is mandatory.

A case in our problem is composed of attributes described in figure 6.

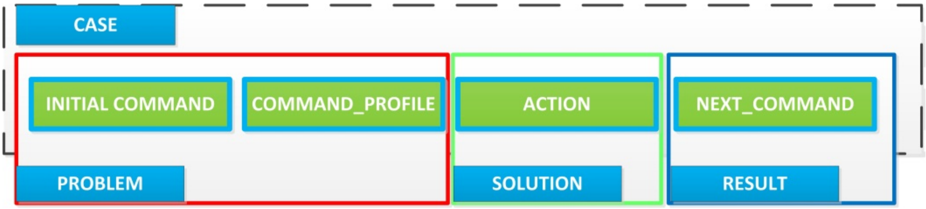


Fig. 6. Definition of CASSHH CBR problem

As you can see, a case comprises the problem, the solution and the result mapped directly on the attributes of our implementation.

What is important to mention is the "Command_profile" attribute that is derived from a previous research of Ramsbrock et. al [15] presented in a scientific paper . In his paper describes the results collected using an SSH honeypot and the way the behavior of an attacker after compromising a SSH account can be modeled as state machine. We have used the seven states described (CheckSW, Install, Download, Run, Password, CheckHW, ChangeConf) and added them manually for each of the 75 Linux commands implemented.

Cases can be viewed, modified and deleted manually by a human expert (during its revision stage). The agent plans (intentions) cannot be generated using different strategies, but in the future it would be feasibly the integration of different algorithms.

8 Conclusions and Future Work

The number of downloaded files on the CASSH honeypot was relatively similar to the ones obtained by the standard Kippo Honeypot but this opens the door for a comparison study. While the CASSHH system generates results by interacting with the attackers without any human intervention, the Kippo Honeypot makes the attackers easily detect is nature. Although the system proposed requires further improvements and more work the initial results are very promising. Another aspect that should be taken into account is the facilitation of the incorporation of new agents using different modeling techniques and learning strategies so that further experiments will allow to compare results obtained by other techniques with these initial results.

9 Annex

Id.	Linux Command	Profile
1	tar	Install
2	grep	CheckSW
3	find	CheckSW
4	ssh	ChangeConf
5	Sed	ChangeConf
6	awk	ChangeConf
7	vim	ChangeConf
8	Diff	CheckSW
9	./xyz	Run
10	export	ChangeConf
11	Xargs	ChangeConf
12	ls	CheckSW
13	Pwd	CheckSW
14	Cd	CheckSW
15	Gzip	Install
16	Unzip	Run
17	paswd	Password
18	ftp	Download
19	crontab	Run
20	service	ChangeConf
21	ps	CheckSW
22	Free	CheckHW
23	Top	CheckSW
24	df	CheckSW
25	kill	ChangeConf
26	rm	Install
27	Cp	Install
28	Mv	Install
29	cat	CheckSW
30	mkdir	Install
31	chmod	Install
32	chown	Install
33	passwd	Install
34	userdel	ChangeConf
35	ifconfig	CheckHW
36	uname	CheckHW
37	id	CheckSW
38	whatis	CheckSW
39	locate	CheckSW
40	history	CheckSW
41	tail	CheckSW
42	less	CheckSW
43	uptime	CheckHW
44	mysql	Run
45	apt-get	CheckSW
46	ping	CheckHW
47	rpm	Install
48	date	CheckSW
49	wget	Download
50	scp	Download
51	fdisk	CheckHW
52	chgrp	Install
53	chroot	Install
54	echo	CheckHW
55	exit	Run
56	jobs	CheckSW
57	lsof	CheckSW
58	nmap	CheckSW
59	netstat	CheckHW
60	nohup	Run
61	dig	CheckSW
62	sudo	Run
63	mount	CheckHW
64	useradd	Password
65	curl	Download
66	perl	Run
67	python	Run
68	gcc	Run
69	make	Run
70	touch	CheckSW
71	w	CheckSW
72	dpkg	Install
73	yum	CheckSW
74	adduser	Password
75	vi	ChangeConf

References

1. Kippo: A ssh honeypot, `http://code.google.com/p/kippo/`
2. Nazario, J.: Kippo Log Analysis, `http://monkey.org/~jose/honeynet/kippo/`
3. The Secure Shell (SSH) Protocol Architecture, `http://www.ietf.org/rfc/rfc4251.txt`
4. Tamminen, U.: Kippo 0.8 released, `http://www.rpg.fi/desaster/blog/2013/04/05/kippo-0-8-released/`
5. Python Twisted, `http://twistedmatrix.com/trac/`
6. In Leake, D. (ed.): Case-Based Reasoning: Experiences, Lessons, and Future Directions. AAAI Press/MIT Press, Menlo Park (1996)
7. Li, L., Tang, W., Wang, R.: A CBR Engine Adapting to IDS. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-M., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005. LNCS (LNAI), vol. 3802, pp. 334–339. Springer, Heidelberg (2005)
8. Zakaria, W., Kiah, M.L.M.: A review on artificial intelligence techniques for developing intelligent honeypot. In: Proceeding of: 8th International Conference on Computing Technology and Information Management, At Seoul, Korea (2012)
9. Burmeister, B., Sundermeyer, K.: Cooperative problem-solving guided by intentions and perception. In: Werner, E., Demazeau, Y. (eds.) Decentralized A.I. 3. North Holland, Amsterdam (1992)
10. Georgeff, M.P., Lansky, A.L.: Procedural knowledge. In: Proceedingsof the IEEE Special Issue on Knowledge Representation, vol. 74, pp. 1383–1398 (1986)
11. Muller, J.P., Pischel, M., Thiel, M.: Modelling reactive behaviour in vertically layered agent architectures. In: Wooldridge, M.J., Jennings, N.R. (eds.) ECAI 1994 and ATAL 1994. LNCS (LNAI), vol. 890, pp. 261–276. Springer, Heidelberg (1995)
12. Shoham, Y.: Agent-oriented programming. Artificial Intelligence 60(1), 51–92 (1993)
13. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, ch. 2, 2nd edn. Prentice Hall, Upper Saddle River (2003) ISBN 0-13-790395-2
14. Laza, R., Gómez, A., Pavón, R., Corchado, J.M.: A Case-Based Reasoning Approach to the Implementation of BDI Agents. In: ECCBR Workshops, pp. 27–30 (2002)
15. Ramsbrock, D., Berthier, R., Cukier, M.: Profiling attacker behavior following SSH compromises. In: DSN 2007: Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pp. 119–124. IEEE Computer Society, Washington, DC (2007)
16. Uses of Class jcolibri.method.retrieve.NNretrieval.similarity.global.Average, `http://gaia.fdi.ucm.es/files/people/juanan/jcolibri/doc/api/jcolibri/method/retrieve/NNretrieval/similarity/global/class-use/Average.html`
17. CASSHH Honeypot code repo: `http://code.google.com/p/casshh/`

Improved Detection of P2P Botnets through Network Behavior Analysis

Shree Garg, Anil K. Sarje, and Sateesh Kumar Peddoju

Indian Institute of Technology Roorkee, Roorkee, Uttrakhand, India
{shreedec,sarjefec,drpskfec}@iitr.ernet.in

Abstract. Botnets are becoming powerful threats on the Internet because they launch targeted attacks towards organizations and the individuals. P2P botnets are resilient and more difficult to detect due to their nature of using different distributed approaches and encryption techniques. Classification based techniques proposed in the literature to detect P2P botnets, report high overall accuracy of the classifier but fail to recognize individual classes at the similar rates. Identification of non-bot traffic is equally important as that of bot classes for the reliability of the classifier. This paper proposes a model to distinguish P2P botnet command and control network traffic from normal traffic at higher rate of both the classes using ensemble of decision trees classifier named Random Forests. Further to optimize the performance, this model also addresses the problem of imbalanced nature of dataset using techniques like downsampling and cost sensitive learning. Performance analysis has been done on the proposed model and evaluation results show that true positive rate for both botnet and legitimate classes are more than 0.99 whereas false positive rate is 0.008.

Keywords: P2P botnets, Random Forests Ensemble, Imbalanced Problem, Network flow, Network Security, Peer to Peer network.

1 Introduction

Botnets are becoming major sources for carrying out various attacks like DDoS, spamming, click-fraud, phishing, and data-stealing [1]. Current generation bots have started communicating through peer-to-peer (P2P) networks and there is no authority control on the files shared on these networks. Recent P2P botnets [2] like Storm, Waledac, and Zeus use distributed Command and Control (C&C) servers and made them more difficult to detect comparing to IRC and HTTP based botnet. Even if one of the C&C servers is detected, it does not disable the entire botnet.

Studies reveal that bots have a specific traffic pattern to communicate over the network, like size of the packets, number of frames transferred in each direction, and duration. So, the network traffic used by botnet C&C channels is distinguishable from normal traffic. Various Data Mining techniques have been applied for intrusion detection as well as botnet detection in the literature. Data mining analysis extracts patterns of attacks, maintain profiles for normal activities and build classifier to detect

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 334–345, 2014.

the attacks [3]. Leveraging the behavior analysis of botnet and normal traffic, this paper proposes a model to detect botnet C&C traffic using ensemble classification with the help of network flow based features.

Rest of the paper is organized as follows: Related work discussed in the literature is given in Section 2. Section 3 explains our proposed approach to detect P2P botnets. Observations and discussion on the results obtained is given in Section 4. Section 5 concludes the paper with suggestion for future work.

2 Related Work

Much of the research has been focused on the detection of botnets. Livadas et al. [4] applied machine learning algorithm namely Naive Bayes, Bayesian network, J48 to identify IRC based botnets. Botminer [5], a protocol independent framework to detect the botnets; they clustered the traffic on the basis of malicious activity and communication pattern of bots. Saad et al. [2] compared Support Vector Machine, Gaussian based, Naive byes, Neural networks to classify P2P botnet traffic on the basis of both host and flow based features. IP addresses and port numbers were also used in the experiments.

Detection and differentiation of IRC, HTTP or P2P bots using host based characteristics is presented in [6]. Some of the works like [2], [7] used the features of P2P traffic based on the behavior of the host and flow. Extraction of host based features is a computationally intensive task as requires network packet level processing [8] and also more storage. BotTrack [9] detect P2P botnets applied PageRank method to host dependency model for tracking communication pattern. It extracts the nodes which are strongly linked to each other. Authors tested their approach on synthetic dataset and to achieve better results it required IP addresses of some bots in the botnet. S. Lin et al. [10] analyzed the differences between P2P botnet behavior and normal traffic on the basis of packet size and packet counts. CoCoSpot [11] detect command and control channels of botnets using message sequence length. C.Haung [12] proposed an approach to detect P2P bots using network failures traffic generated by bots. In our previous work [13], different machine learning algorithms like Nearest Neighbor (NN), Naive Bayes, and J48 have been analyzed, for the detection of P2P botnets using various network traffic features. The results indicate that the accuracy of the classifier trained using NN and J48 is high. However the detection of non-bot class is very poor. It is understood that each algorithm has some limitations in terms of testing time or inter-class classification. There is a need for designing a model that can detect P2P botnet C&C traffic independent of parameters like payload, port numbers, IP address, and host features based analysis. In addition it should not require the prior knowledge of bot host in the monitored network and for the high confidence in the result a classifier should have high sensitivity and high specificity.

3 Proposed Approach

In this section we present our proposed approach for detection of bot traffic. The main focus of the proposed model is to detect P2P botnets based on the traffic analysis.

The key feature of this model is to detect bots without any prior knowledge of bot hosts in the network and is also independent of IP addresses, port numbers, payload, and host features. The phases involved in the proposed model are diagrammatically represented in Figure 1. We examine the behavior of botnet and non-botnet (P2P application and normal environment) traffic at the level of the TCP/UDP flows. Details of experimental set up, dataset and analysis of the traffic is given in Behavior Analysis Section 3.1. On the selected features and classifier, experiments are performed to detect the botnet traffic. Classifier used to detect the botnets, its training and testing is given in Section 3.2. The issue of imbalanced dataset is also addressed and their details are given in Section 3.3.

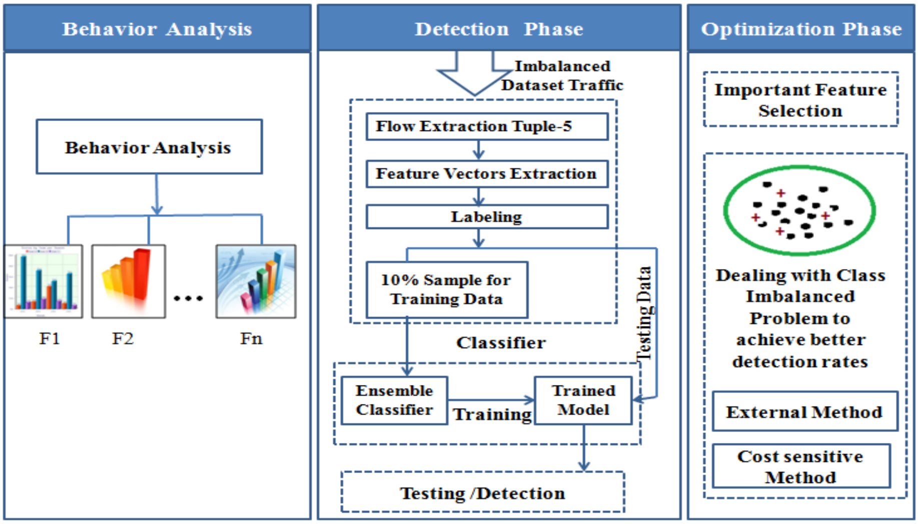


Fig. 1. Model of the proposed approach

3.1 Behavior Analysis

The behavior of bots is mostly dependant on various packet related parameters like size, duration, number, ratio or average values [2], [6], [7] and [10]. Incoming or outgoing bytes per flow and the number of frames per flow may help to understand the nature of bot traffic better. Generally a typical transaction of normal traffic is small request and large response, or a quick ping-pong setup with either a large upload or large download to/from the client. Moreover bots use small size of packets [10]. Features extraction can be based on host and/or flow properties of the network traffic. Host based network features require analysis of each packet belonging to a particular host and consume huge time [15]. Extraction of flow based features takes considerable lesser amounts of time and space as we have to analyze only flow statistics.

Selection of features is important to distinguish network traffic of different applications. To analyze the behavior, we have considered the features tabulated in Table 1 to distinguish between the normal and botnet traffic.

Table 1. List of selected features

Features	**Explanation**
Iop_byte	Ratio of incoming over output bytes in the flow
Apl	Average packet length in the flow
Tbf	Total bytes transferred during whole capture to the number of frames per flow
Framespersec	Frames per second in the flow
Bytespersec	Bytes per second in the flow
Iop_frame	Ratio of incoming over outgoing number of frame in the flow
Duration	Duration of the flow
Fromframe	Number of Incoming frames in the flow
Toframe	Number of outgoing frames in the flow
Payload	Payload size of all the packets in the flow
Tobyte	Number of outgoing bytes in the flow
Frombyte	Number of incoming bytes in the flow

Experiments are performed to analyze the behavior of network flow traffic for botnet and non-bot traffic. The experimental setup and dataset used for all the experiments is discussed below:

Experimental Setup: HP Z600 Workstation is being used during the course of experiments. UDP and TCP flows are extracted using the utility of Tshark[1] and a Perl script to compute features of the extracted flows from the dataset. Traffic is labeled into two classes' bot and non-bot.

Dataset: For all the experiments, dataset has been taken from ISOT research Lab, University of Victoria [2]. Dataset is the combination of several existing bot and non-bot data. Bot traffic is the combination of Storm, Waledac and Zeus botnets whereas non-bot traffic is a mixture of a variety of applications, P2P protocols (Bit-Torrent, eDonkey, Gnutella, DirectConnect; VoIP), chat applications (Skype, MSN Live; FTP sessions, file transfer with download manager; e-mail sending, receiving sessions) and web traffic (e-mail, SSH sessions, SCP sessions, FPS, MMORPG gaming sessions, streaming radio; streaming video and web based streaming and enterprise network traffic). There are total 667413 flows extracted from this dataset.

In this work, network flow is defined as tuple-5: Source IP address, Source port number, Destination IP address, Destination port number and protocol (TCP/UDP) which have transferred at least one packet in both directions. Figure 2 indicates the behavior of network flow traffic for botnet and non-bot traffic classes.

[1] `http://www.wireshark.org/docs/man-pages/tshark.html`

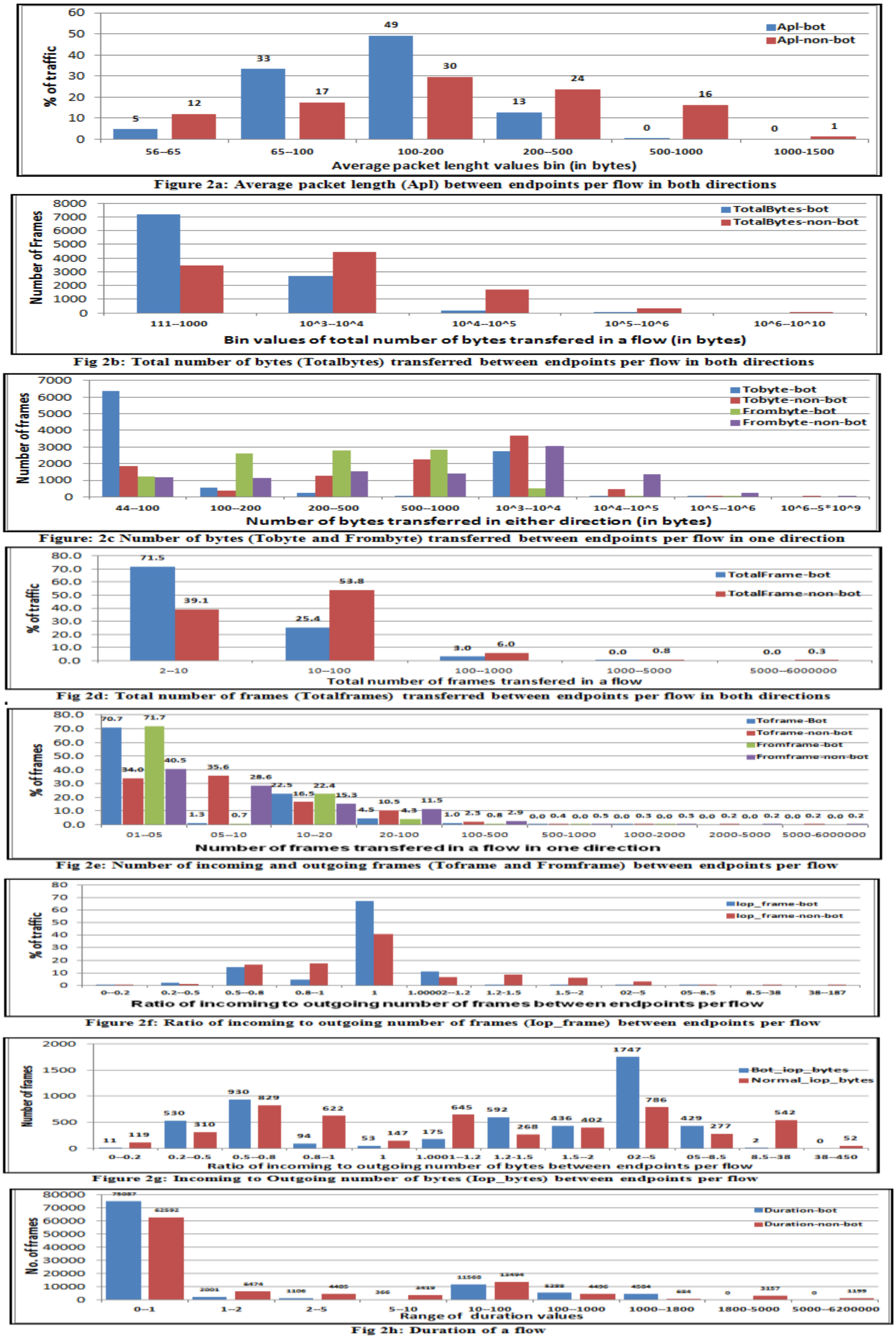


Figure 2a: Average packet length (Apl) between endpoints per flow in both directions

Fig 2b: Total number of bytes (Totalbytes) transferred between endpoints per flow in both directions

Figure: 2c Number of bytes (Tobyte and Frombyte) transferred between endpoints per flow in one direction

Fig 2d: Total number of frames (Totalframes) transferred between endpoints per flow in both directions

Fig 2e: Number of incoming and outgoing frames (Toframe and Fromframe) between endpoints per flow

Figure 2f: Ratio of incoming to outgoing number of frames (Iop_frame) between endpoints per flow

Figure 2g: Incoming to Outgoing number of bytes (Iop_bytes) between endpoints per flow

Fig 2h: Duration of a flow

Fig. 2. Traffic behaviour of (a)Apl (b)Totalbyte (c)Tobyte and Frombyte (d)Totalframe (e)Toframe and Fromframe (f)Iop_frame (g)Iop_byte (h)Duration (i)Framepersec (j)Bytepersec (k)Tbf features

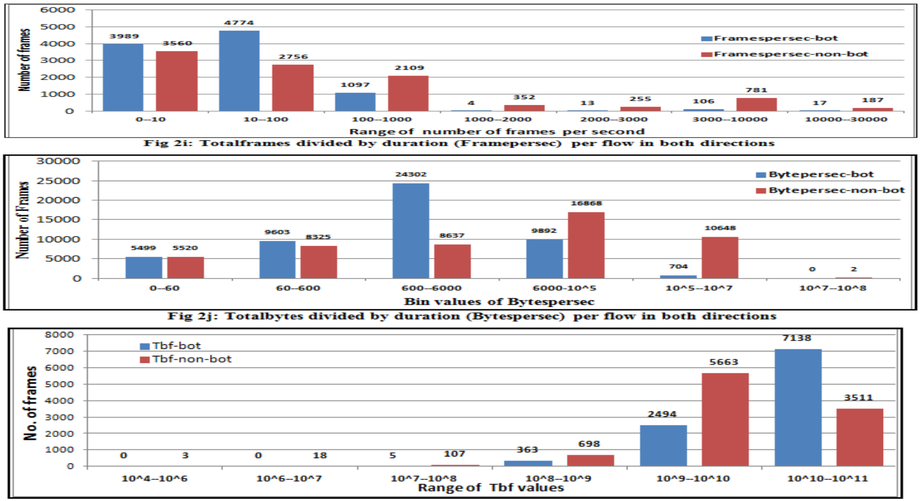


Fig 2i: Totalframes divided by duration (Framepersec) per flow in both directions

Fig 2j: Totalbytes divided by duration (Bytespersec) per flow in both directions

Fig 2k: Total bytes by frames (Tbf) per flow in both directions

Fig. 2. (*Continued*)

The IP packets in the traffic have different lengths. It has been observed (Fig: 2a) that bot traffic uses small size of packets as well as have less number of distinct packet sizes in comparison to non-bot traffic. Bots traffic ranges from 56 bytes to 905 bytes having 307 distinct values whereas benign traffic ranges from 56 bytes to 1471 bytes having 1301 distinct values. Less number of Totalbytes has been transferred between bots in comparison to non-bot end points (IP address and port number) in a flow as bot keeps on changing the port number to make new connections (Fig: 2b). As shown in Fig 2c, bots receives comparatively small size of packets to the size of packets sent whereas it is opposite in case of normal.

Bot maintains connections but transfers only a small number of frames (Fig. 2d) to remain hidden using less bandwidth on the network which is not true in case of normal traffic. Toframe and Fromframe have a relation as shown in Fig. 2e. Normal traffic is a mixture of traffic from different source like P2P application, Web server and, chat. Fig. 2f indicates that the bot traffic contribution is high when Iop_frame ratio goes close to unity similarly as shown in Fig. 2g, mostly. Iop_byte value for bot traffic lies between 2 to 5 which means command may be of large size in comparison to response or it may be bot updates, new malware, spam templates, and email lists. Fig 2h indicates that, normal traffic duration values are distributed over a wide range while in case of bot traffic it is limited up to close to 100 sec. Fig: 2i and 2j indicate Framespersec ratio and Bytespersec ratio are also small for bots. Tbf will be comparatively high to that of non-bot traffic as shown in Fig: 2k.

3.2 Detection Phase

The main focus of this work is to detect the botnet flows and detection is highly dependent on performance of the classifier. The performance evaluation can be done using various parameters like True Positives Rate (TPR: TP / TP+FN), True Negative Rate (TNR: TN / TN + FP) and False Positives rate (FPR: FP / FP+TN) of the classifier where TP: botnet traffic predicted as botnet, TN: non-bot traffic predicted as non-bot, FP: non-bot traffic predicted as botnet. If TPR and TNR of a binary classifier are high, its overall accuracy will be high.

Classifier: There are many classifiers available based on data mining techniques. Ensemble methods are used to achieve better prediction performance than that could be obtained from single model. Random Forests (RF) is an ensemble classification or regression approach, unsurpassable in accuracy among current data mining algorithms [14]. When compared with other ensemble based classifiers (for eg. Adaboost), RF is "favorably" comparable, only that RF is cheaper in terms of computing time and more robust to the noise data. RF is computationally efficient technique which can operate large dataset quickly. RF is a combination of many decision trees using random feature selection. Each decision tree takes little amount of time to build a model but it takes negligible time for testing as it works on rules to classify the instances. RF is also suitable for learning imbalanced [15] problem as class weights are incorporated in it. Malicious traffic (bot) is always very less in comparison to normal traffic. Hence, choosing RF is justifiable.

For the detection, classifying model of RF is trained on the metrics selected from the network flows listed in Table1. Then testing is done on the features vectors extracted from the dataset for the detection of botnets.

Training: For selecting the training sample, only 10% out of total flows is randomly chosen using Resample method. It produces a random subsample of a dataset using sampling with replacement.

Testing and Detection: Entire dataset is used for testing. The accuracy of RF depends on the strength of each tree in the forest. But high correlation between any two trees increases the error rate of classifier. Increasing (or decreasing) randomly chosen number of features (M) to split a node in tree increases (or decreases) both the correlation and the strength. On the other hand, number of trees in the forest (N) also plays an important role for time complexity and strength of classifier. Experiments have been performed for different values of M ranging from 12 to 5 (12, 10, 8, 7, 6, 5) with values of N being 10, 50 and 100. An experiment with 500 trees is also performed and results are improved as expected but the time complexity is comparatively very high so, the results are discarded.

From Figure 3, it is observed that when M increases keeping N constant, TPR first improves reaching optimal value at M=8 and then decrease slightly. When we increase N keeping M constant, TPR improves while TNR does not change much. The dashed line across the graph indicates the values of N=100 and M=8, from where there is no significant improvement in the performance. This indicates that using more or less than 8 features to split a node in the tree decreases the performance of classifier. For rest of the experiments the values of M and N are fixed as 8 and 100 respectively.

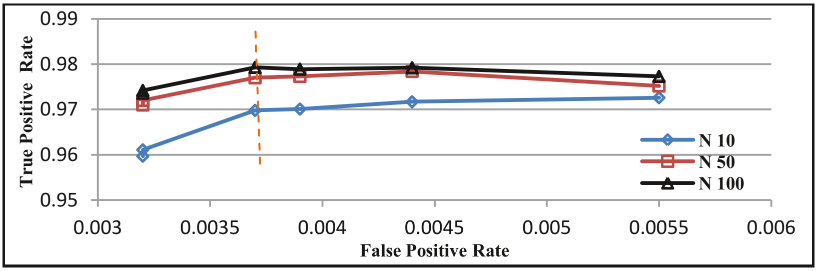


Fig. 3. ROC graph by decreasing the value of M (to split a node in the tree) from 12 to 5

3.3 Performance Optimization

This paper adapts various steps to optimize the performance of the algorithm. The steps used in this paper are discussed below.

Step 1. Important Features Selection: In order to improve the detection interval, some of the important features can be selected instead of all features. We have chosen the features which are more contributing in learning than others by evaluating the importance of each feature using feature selection algorithm provided by RF. To evaluate the importance of the attributes, out-of-bag cases (data left by training sampling) are used to count the number of votes for the correct class first, for every tree grown in the forest second, by randomly permuting the values of variables. Then the difference of the number of votes for the correct class divided by the number of trees in the forest is *variable importance* score for variables.

Figure 4 plots a graph for variable importance of each feature in the decreasing order of their importance. Iop_byte, Apl, Tbf, Framespersec are more important attributes, as predicted by the algorithm. Results indicate that selecting only top 8 more important features gives almost same result as that with all the features.

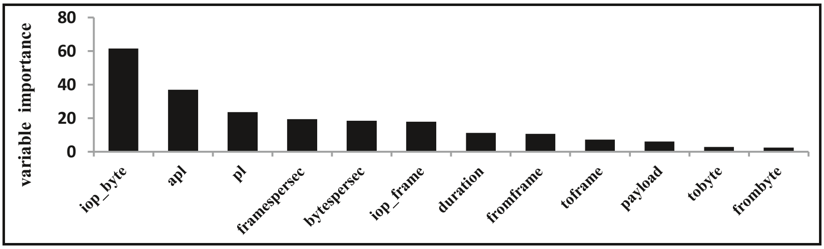


Fig. 4. Variable importance of the features

RF reduces the overall error rate by reducing the error rate of majority class while increasing the error of minority class [14]. In contrast, we have focused to improve the detection rate of botnet class keeping detection rate of non-malicious class high. This issue can be resolved by addressing the class imbalance problem in the dataset.

Step 2. Addressing Imbalanced Nature of Data: Imbalance problem is one of the major challenges in data mining community. Imbalanced data has higher number of instances of one class (majority class) and comparatively very less number of instances of other class (minority class. In our dataset the number of flows for both classes is not equal. Non-Malicious traffic contributes only *43475* flows in the traffic while non-malicious flows dominate the rest of traffic. It is natural in real time environment because attack (bot) traffic is always less in comparison to normal. In Data mining many techniques have been developed to address this problem. According to M. Galar et al. [16] these techniques can be categorized into three groups: Internal, External and Cost sensitive methods.

However, to improve the results of imbalanced dataset we have performed our experiment using two methods: a) External method b) Cost sensitive method.

a) External Method (Data level Changes): Dataset can be balanced using either DownSampling or OverSampling methods. However we do not find it suitable to use OverSampling method here since it replicates minority class instances. Synthetic bot flow can mislead the model. In this method, DownSampling has been done over this dataset. The number of instances of majority classes is reduced while keeping the minority class instances as it is by imbalanced dataset. From the imbalanced dataset, there is 62452 and 4289 number of non-bot flows and botnet flows respectively in our training sample. The instances of non-bot traffic is downsampled to 80%, 73%, 66%, 60%, 57%, 54% and 50% using a Perl script in order to balance the number of instances of both the classes and randomness in the dataset (Figure 5). As we keep on decreasing the non-bot instances in the training data it deteriorates TNR. TPR improves till (shown with dashed line in the graph) we have 60% not bot traffic and 40% bot traffic. After that bot traffic detection rate does not change much but the results of non-bot class falls significantly hence raise more false alarms.

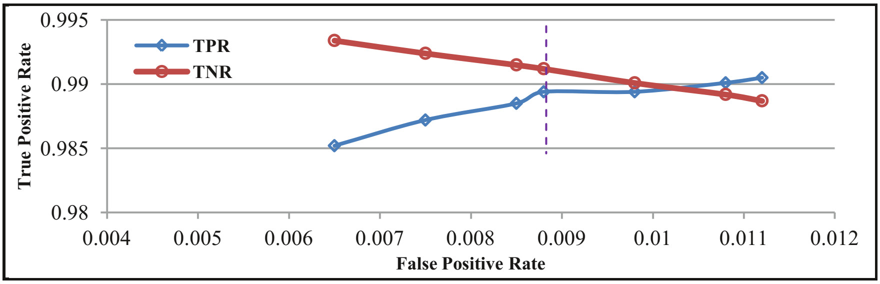


Fig. 5. Performance by downsampling the majority class in increasing order

DownSampling is working equally well for both cases and true positives reached nearly 99% and FPR 0.008.

b) Cost Sensitive method: It integrates both data level changes (by adding costs to instances) and algorithm level modifications (by modifying the learning process to accept costs).

Adding Weights: More weight is given to the minority class in order to penalize high penalty for misclassify minority class instance. Training sample remains same as for imbalanced dataset.

Algorithm Changes: RF incorporates weights into it, in tree induction process and terminal nodes of each tree. The prediction is calculated using weighted majority votes.

Parameters are chosen as M=8 and N=100 for top 8 features. Weights ratio used in the experiment are Non-Bot: Bot 1:1, 1:2, 1:3, 1:5, 1:10. The obtained results indicate that higher the weights assigned to minority class increase both the detection rate and false alarms (Figure 6). Results obtained for 1:10 ratio outperforms for botnet class but it decreases the TNR significantly. Experiments show, by assigning higher weights the results for minority (bot) class is improved significantly but at the cost of high false positives. More than 99% of TPR and TNR are achieved with 1:5 weights ratio (dashed line in the graph) to non-bot and bot classes with FPR less than 0.0085.

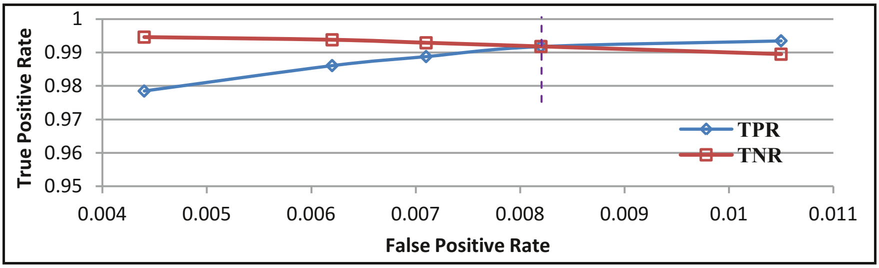


Fig. 6. Performance by assigning the higher weights to the minority class in increasing order

4 Discussion of the Results

For original (imbalanced) dataset TPR is 97.93%. This has been further optimized by handling the imbalanced nature of the dataset using downsampling and cost sensitive methods. So as per observation, downsampling method raises more false alarms in comparison to that of by weights assignment method because in downsampling method less training sample of non-bot class was used to train the model. Thus we have obtained more than 99% of true positive for both the classes (bot and non-bot) keeping false alarm as low as 0.0082 (Table 2).

Table 2. Summarized results (Keeping M=8, N=100 with only Top 8 features)

	Original dataset	Downsampled dataset	Original dataset (1 : 5)
Bot Detection	0.9793	0.9895	0.9918
Non-bot Detection	0.9962	0.9914	0.9918
FPR	0.0038	0.0086	0.0082
Overall Accuracy	0.9950	0.9912	0.9918

Table 3 contains the comparison of our obtained results with the other published work based on network traffic using classification to detect P2P botnets. The proposed method classifies more than 99% of botnet as well as non-botnet traffic correctly while other does not do so, overall accuracy of the classifier does not always mean the accuracy of both the classes.

Table 3. Comparison with other published work

Paper	P2P Bots	TPR %	TNR %	Accuracy %	FPR %
G. Fedynyshyn et al. [6]	2	99.2	92.17	92.9	7.8
P.Barthakur et al. [17]	1	99.7	89	99	11
S.C. Lin et al. [10]	1	-	-	98	-
S.Garg et al. [13]	3	89	99	> 99	13.4
Proposed Model	3	99.18	99.18	99.18	0.8

Evaluation with different datasets may lead to different results so to cover a wide range of P2P botnets, a dataset with three classes of botnets is used and non-bot traffic is also mix of several applications. We believe that evaluation of the proposed approach on three real P2P botnets traces would be more reliable.

5 Conclusion

P2P botnets are growing fast and also difficult to neutralize as they are controlled via distributed C&C servers. In order to achieve accurate detection, this paper proposes a model for P2P botnet detection with different flow based features. A thorough analysis is done in selecting suitable flow based features of the traffic. This study infers that Average packet length, Iop_byte, Iop_frame and Totalframes are useful features to understand the behavior of botnet traffic. Further, the parameters of RF algorithm are fine tuned for better detection of bots. This work has also focused on command and control traffic to identify the presence of botnet before they launch any attacks. The results of the experiments indicate that there is a considerable increase in true positive rate of both classes independently while keeping false positive rate very low. However, these experiments are performed on a fixed dataset. In future, this work aims to target the real time detection while keeping possibility of attacker mimicking to bypass the detection.

References

1. Chen, L., Richard, R.B.: Timing analysis in P2P botnet traffic using probabilistic context-free grammars. In: CSIIRW 2013. ACM, Oak Ridge (2013)
2. Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Wei, L., Felix, J., Hakimian, P.: Detecting P2P botnets through network behavior analysis and machine learning. In: 2011 Ninth Annual International Conference on Privacy, Security and Trust (PST), pp. 174–180 (2012)

3. Hand, D., Mannila, H., Smyth, P.: Principles of Data Mining. MIT Press, Cambridge (2001)
4. Livadas, C., Walsh, R., Lapsley, D., Strayer, W.T.: Usilng Machine Learning Technliques to Identify Botnet Traffic. In: Proceedings 2006 31st IEEE Conference on Local Computer Networks, pp. 967–974 (2006)
5. Guofei, G., Roberto, P., Junjie, Z., Wenke, L.: BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection. In: Proceedings of the 17th Conference on Security Symposium. USENIX Association, San Jose (2008)
6. Fedynyshyn, G., Chuah, M.C., Tan, G.: Detection and classification of different botnet C&C channels. In: Calero, J.M.A., Yang, L.T., Mármol, F.G., García Villalba, L.J., Li, A.X., Wang, Y. (eds.) ATC 2011. LNCS, vol. 6906, pp. 228–242. Springer, Heidelberg (2011)
7. Junjie, Z., Perdisci, R., Wenke, L., Sarfraz, U., Xiapu, L.: Detecting stealthy P2P botnets using statistical traffic fingerprints. In: 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN), pp. 121–132 (2011)
8. Alaidaros, H., Mahmuddin, M., Mazari, A.A.: An Overview of Flow-based and Packet-based Intrusion Detection Performance in High Speed Networks. In: Proceedings of the International Arab Conference on Information Technology (2011)
9. François, J., Wang, S., State, R., Engel, T.: BotTrack: Tracking Botnets Using NetFlow and PageRank. In: Domingo-Pascual, J., Manzoni, P., Palazzo, S., Pont, A., Scoglio, C. (eds.) NETWORKING 2011, Part I. LNCS, vol. 6640, pp. 1–14. Springer, Heidelberg (2011)
10. Lin, S.-C., Chen, P., Chang, C.-C.: A novel method of mining network flow to detect P2P botnets. In: Peer-to-Peer Networking and Applications, pp. 1–10 (2012)
11. Dietrich, C.J., Rossow, C., Pohlmann, N.: CoCoSpot: Clustering and recognizing botnet command and control channels using traffic analysis. Comput. Netw. 57, 475–486 (2013)
12. Huang, C.-Y.: Effective bot host detection based on network failure models. Computer Networks 57, 514–525 (2013)
13. Garg, S., Singh, A.K., Sarje, A.K., Peddoju, S.K.: Behaviour Analysis of Machine Learning Algorithms for detecting P2P Botnets. In: International Conference on Advanced Computing Technologies (2013)
14. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)
15. Chao, C., Andy, L., Leo, B.: Using Random Forest to Learn Imbalanced Data. University of California, Berkeley (2004)
16. Galar, M., Ferna, X., Ndez, A., Barrenechea, E., Bustince, H., Herrera, F.: A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42, 463–484 (2012)
17. Barthakur, P., Dahal, M., Ghose, M.K.: A Framework for P2P Botnet Detection Using SVM. In: Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 195–200 (2012)

SLASS: Secure Login against Shoulder Surfing

Nilesh Chakraborty and Samrat Mondal

Computer Science and Engineering Department
Indian Institute of Technology Patna
Patna-800013, Bihar, India
{nilesh.pcs13,samrat}@iitp.ac.in

Abstract. Classical password based schemes are widely used because it provides fair security and yet easy to use. However, when used in a public domain it is vulnerable to shoulder surfing attack in which an attacker can record the entire login session and may get the user's original password. To avoid such attack, we have proposed a methodology known as Secure Login Against Shoulder Surfing or SLASS which is based on a partially observable attack model where an attacker can partially observe the login session. In the proposed scheme, the attacker cannot see or hear the challenges thrown by the system but can only see the responses provided by the user. User remembers a password of five characters long consisting of alphabets only and the responses are provided by some directional keys. Experimental analysis show that our scheme is less error prone, easy to use and provides high security compared to some existing approaches.

Keywords: Authentication, Password, Shoulder Surfing, Partially Observable.

1 Introduction

Authentication is a very important aspect during human interaction with system. A secured and reliable system can always help to differentiate between an attacker and a legitimate user. Password based authentication [5] is one of the most popular authentication scheme that strengthens the reliability of the system against different types of attacks. With the increase in security in the authentication techniques, new attack models have also been developed [1] [3]. One of such attack that compromises the security of classical password entry mechanism is *shoulder surfing attack* where an attacker can record the entire login session of an user and get to know the user actual password. Shoulder surfing attack from the attacker perspective are mainly of two types - a) *Weak Shoulder Surfing attack* where an attacker is a human with limited cognitive skill [11] and b) *Strong Shoulder Surfing attack* where an attacker uses some machinery (like recording device) [7] to record entire session. Our proposed methodology can be used to resist both kind of attacks.

Many methodologies [10],[11], [8] have been proposed to resist shoulder surfing attack. Using those methodologies the system will generate different session

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 346–357, 2014.

passwords for each session on the basis of user's original password. As the session password varies depending upon the user's original password and as user does not reveal his actual password in any session so only by recording sessions, an attacker may not get the actual password. While some of these shoulder surfing resistance methodologies have been implemented on partially observable system [10] some are implemented on fully observable system [16]. Our proposed methodology SLASS implemented on a partially observable system where an attacker can observe the partial input output operation performed by users during a session. The user gives response to the challenges generated by system. User will listen those challenges through some protected media so that attacker does not get access to those challenges. The media must not be compromised against any kinds of side channel attack [6][17]. We have proposed SLASS motivated by SSSL [10] technique where an user uses different directions to give his response. We use some of those directions as a *challenge* generated by the system and come up with some better security compared to SSSL in terms of password guessing attack.

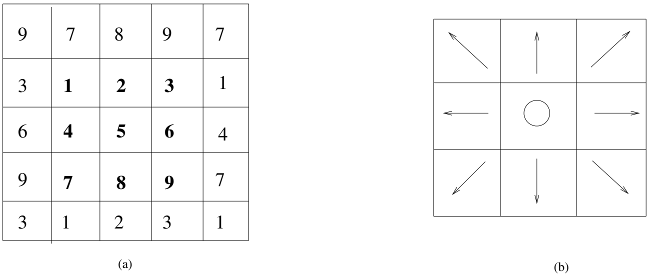
The rest of the paper is organized as follows. In Section 2 we will briefly discuss SSSL technique and show some of the drawbacks of those techniques. Our proposed SLASS scheme is discussed in Section 3. Important features and usability evaluation during login is discussed in Section 4. We have compared SLASS with some other existing methodologies in Section 5. Finally we conclude and give future direction to our work in Section 6.

2 Motivation

Two broad categories of passwords are – (a) Graphics based password and (b) Text based password. Influenced by the fact that human can easily remember picture than text [2] many graphics based password have been proposed. Some of these methodologies – like PassFace [9], PassGo [12], PassPoint [14] S3PAS [16], SSSL [10] were developed to resist shoulder surfing attack. The way user interacts with the system in all these schemes are different in nature. Among these, SSSL uses direction based response and that does not require any prerequisite knowledge. Our proposed SLASS scheme is a text based password scheme which has been motivated from the concept of the directional key response of SSSL.

2.1 A Brief Overview of SSSL

Shoulder Surfing Safe Login (or SSSL) [10] was proposed in the year of 2009. In this scheme user remembers 5 digits from the set $\{1, 2, ..., 9\}$. The table used in SSSL method constructed such a way that every digit i is an immediate neighbour to other 8 digits from the set $\{1, 2, ...9\}$. Challenges come to the user in 5 rounds during a session in SSSL. Challenge values are pseudo random [4] and it is generated from the set $\{1, 2, ..., 9\}$ one at a time. After receiving the challenge value user will first locate the original PIN digit from the bold part of the table as shown in Fig.1(a). Once the digit is located then user finds the challenge digit from its neighboring cells. Depending upon the position of the original PIN digit with respect to challenge digit, the user enters appropriate direction from the keypad which is shown in Fig. 1(b).



9	7	8	9	7
3	**1**	**2**	**3**	1
6	**4**	**5**	**6**	4
9	**7**	**8**	**9**	7
3	1	2	3	1

Fig. 1. (a)Orientation of digits (b) Keypad structure for SSSL

The following example will give a clear idea about how SSSL methodology works. Suppose user has chosen a five digit password 96978. Then corresponding to a challenge value 36571, a valid user response has been shown in Table 1. As digit 3 is placed under the value 9 in Fig.1(a) so the user will press the down arrow. The rest of the responses of Table 1 can also be obtained in a similar manner. Though this methodology works fine against shoulder surfing attack but it has a limitation as SSSL does not include digit 0 in it and uses only 9 directions. So probability of guessing the password in this 5 rounds of challenge response scheme is $1/9^5$. Another drawback of this methodology is the co-relation relationship between digits are identifiable. Every co-relation between digits decrease the security factor of SSSL by 1/9.

Table 1. User response table of SSSL

password	Challenge	Response
9	3	$\downarrow$
6	6	$\circ$
9	5	$\nwarrow$
7	7	$\circ$
8	1	$\swarrow$

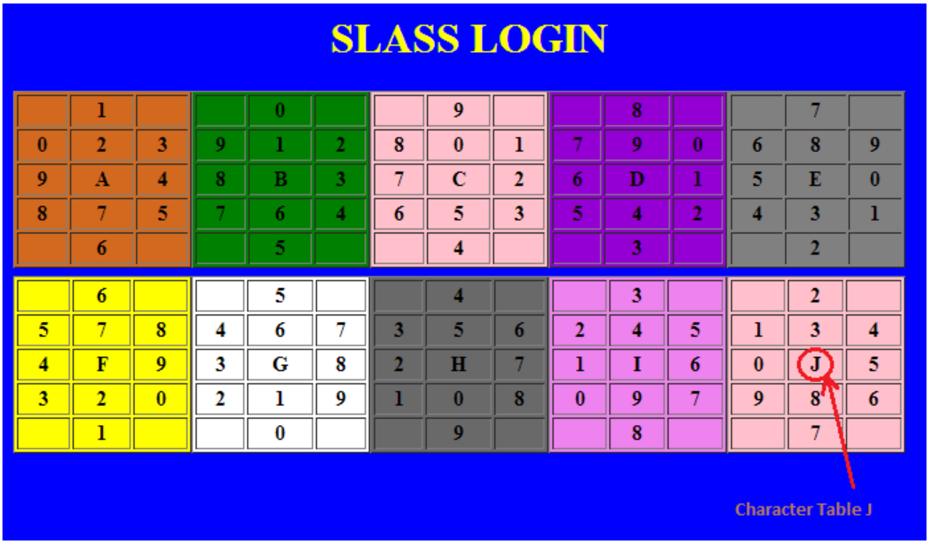
In SLASS we have overcome this two drawbacks of SSSL. In SLASS we make use of digit 0 as a valid response. So probability of guessing the password has been increased to $1/10^5$. SLASS avoids identifying co-relations (or similarities) between digits by an attacker. In the next section we will give a broad overview of our proposed methodology.

3 Proposed Methodology

SLASS is a 5 round challenge response scheme. During login process user will give responses to the five challenge values generated by the system on the basis of his actual password. User will choose his password from a set of 10 characters $\{A, B, ...J\}$. The password might contain repetitive characters. For example $AABCD$ might be a valid user password. System will generate 10 character tables as shown in Fig. 2. Each table is known by the character placed at center position of the table. In Fig. 2, *Table J* is marked, the other tables can also be marked in a similar manner. All the 10 digits from the set $\{0, 1, ..., 9\}$ are surrounded by the center character of the table. The important features of the character tables are – if any digit appears in the p^{th} position in character *Table* T_i then same digit will not appear at p^{th} position of any other character table. For example, in *Table A* digit 9 appears at the left position of the center character and digit 9 has not appeared at the left position with respect to center character in any other character tables.

3.1 User Interface in SLASS

All the ten character tables are organized in a screen so that user can easily locate the character in each table. Also different colors are used to distinguish each table from the other. Once user provides his username then the screen containing ten character tables will appear to the user. The interface then provides challenges using some secured audio media like ear phone. Thus only the user can listen to the challenge. After listening to the challenge, user will move to his chosen



SLASS LOGIN

	1			0			9			8			7	
0	2	3	9	1	2	8	0	1	7	9	0	6	8	9
9	A	4	8	B	3	7	C	2	6	D	1	5	E	0
8	7	5	7	6	4	6	5	3	5	4	2	4	3	1
	6			5			4			3			2	

	6			5			4			3			2	
5	7	8	4	6	7	3	5	6	2	4	5	1	3	4
4	F	9	3	G	8	2	H	7	1	I	6	0	J	5
3	2	0	2	1	9	1	0	8	0	9	7	9	8	6
	1			0			9			8			7	

Character Table J

Fig. 2. User Interface for SLASS

character table and will give response accordingly using the keyboard shown in Fig. 3. We keep the structure of keypad similar to the classical password entry keypad with which the user is already familiar. The *cancel* button in user keyboard will abort a session and *reset* button will start a new session.

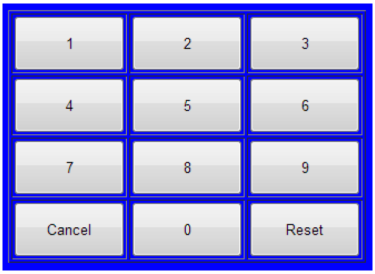


Fig. 3. User Login keypad

3.2 User Login Procedure

During login user will interact with the system in the following manner –

- User will first enter his username to the system.
- Then system will generate SLASS tables using Algorithm 1
- User will give response to all the 5 challenges in a session on the basis of his chosen password.
- User responses will be evaluated using Algorithm 2
- After evaluating user response, system will decide user authenticity using Algorithm 3.

To generate the character tables we have used Algorithm 1. The algorithm uses two arrays *allchar* and *num*. The array *allchar* holds characters A to J in the alphabetical order and the contents of *num* array is shown in Table 2. Index 5 of the array is set with some arbitrary value (here 100) as this index will not be used to build tables in SLASS.

Table 2. Content of array num

Index	content	Index	content
0	1	6	4
1	0	7	8
2	2	8	7
3	3	9	5
4	9	10	6
5	100	–	–

Algorithm 1. Generating tables in SLASS

```
Input: This algorithm will take array allchar [A,B,...J] and num [1, 0, 2, 3, 9, 100,
4, 8, 7, 5, 6] as input.
Output: This algorithm will generate 10 different character table for SLASS
Initialize: $index \leftarrow 0$
for i = 10 to 1 do
  for j = 0 to 10 do
    $K \leftarrow (i+$ num[$j$]) mod 10
    if $j \neq 5$ then
      $Table_{allchar[index]}$. CELL(j).Value ← K
    else
      r := $10 - i$
      $Table_{allchar[index]}$. CELL(j).Value ← allchar[r]
    end if
  end for
  index++
end for
```

$Table_A.CELL(j)$ denotes j^{th} cell in character Table A. In the above algorithm for index value 0, *allchar[index]* will hold the value A.

User will receive challenges through a trusted medium. Challenges will come in terms of directions to the user. Challenge values will be comprised of the direction from the set {*Left*, *Right*, *Up*, *Down*, *Left-up-diagonal*, *Right-up-diagonal*, *Left-down-diagonal*, *Right-down-diagonal*, *Super-up*, *Super-down*}. For convenient we abbreviated *Left* as *L*, *Right* as *R*, *Up* as *U*, *Down* as *D*, *Left-up-diagonal* as *LUD*, *Right-up-diagonal* as *RUD*, *Left-down-diagonal* as *LDD*, *Right-down-diagonal* as *RDD*, *Super-up* as *SU* and *Super-down* as *SD*.

All these directions are with respect to the character placed at the center of the table. At a particular session a user will listen five challenge values and will response accordingly to authenticate himself. A particular challenge might come more than once. We have stored this sounds to an array *audio@challenge* (indexes ranges from 0 to 9). The relative positions with respect to center has shown in Table 3 and the content of array *audio@challenge* is given in Table 4 respectively.

During a session the 5 indexes of the array will be chosen randomly and the sound at that index will be carried through a secure media to the user. Same index might be chosen more than once.

3.3 Methodology for Giving Response in SLASS

In this section we will discuss about password entry mechanism performed by user in SLASS. User will first select his 5 character long password from the

Table 3. Direction with respect to Char cell. In Table 4, index of array denotes cell number of this table and sounds of direction denotes corresponding direction with respect to Char.

	1	
0	2	3
9	*Char*	4
8	7	5
	6	

Table 4. Array audio@challenge contains sounds of different directions

Index of array	Sounds of direction
0	*Left-up-diagonal*
1	*Super-up*
2	*Up*
3	*Right-up-diagonal*
4	*Right*
5	*Right-down-diagonal*
6	*Super-down*
7	*Down*
8	*Left-down-diagonal*
9	*Left*

set $\{A, B, ...J\}$. Suppose user chosen password is *AJICI.* Now while going for login, user will listen 5 random challenge values against each character of his password. If user listens his first challenge as *Left* then corresponding to his first secret password character A he will go to the character *Table A* and will enter the digit 9 placed at the left with respect to center character A of that table. If the second challenge is received as *Super Up* then user will go to the character *Table J* (user's second character of the password) and will enter the digit placed at above up (or *Super UP*) position with respect to center character of *Table J.* Here the corresponding digit will be 2. The demonstration of user response for user password $AJICI$ for a particular session has been given in the Table 5 and in Table 6 the response is shown for a different session.

Table 5. User response table of for password AJICI

password	Challenge	Response
A	*SU*	1
J	*LUD*	1
I	*R*	6
C	*LUD*	8
I	*SD*	8

Table 6. User response in different session for password AJICI

password	Challenge	Response
A	*SD*	6
J	*L*	0
I	*R*	6
C	*RDD*	3
I	*U*	4

System will evaluate the user response using Algorithm 2. User will choose 5 characters form set $\{A, B, ..., J\}$ which will be stored in array *usechar* (index ranges from 1 to 5) and user's response will be captured in array *click* (index ranges from 1 to 5).

Algorithm 2. Evaluating user response in SLASS

```
Input: This algorithm will take array click[i] as input
Output: Will update array X[i]
for i = 1 to 5 do
   for k = 0 to 9 do
      if (allchar[k] = usechar[i]) then
         if (usechar[i] = ‘A’) || (usechar[i] = ‘F’) then
            K = (K + Listensound[i]) mod 10
            if click[i] = k then
               X[i] = 1
               break
            end if
         end if
         if (usechar[i] = ‘B’ || usechar[i] = ‘G’) then
            K = (K + Listensound[i] − 2) mod 10
            if click[i] = k then
               X[i] = 1
               break
            end if
         end if
         if (usechar[i] = ‘C’ || usechar[i] = ‘H’) then
            K = (K + Listensound[i] + 6) mod 10
            if click[i] = k then
               X[i] = 1
               break
            end if
         end if
         if (usechar[i] = ‘D’ || usechar[i] = ‘I’) then
            K = (K + Listensound[i] + 4) mod 10
            if click[i] = k then
               X[i] = 1
               break
            end if
         end if
         if (usechar[i] = ‘E’ || usechar[i] = ‘J’) then
            K = (K + Listensound[i] + 2) mod 10
            if click[i] = k then
               X[i] = 1
               break
            end if
         end if
      end if
   end for
end for
```

User response will be evaluated by the array X (index ranges from 1 to 5). *Listensound* is another array that we have used to store those indexes of array *audio@challenge* which has been used by the system, for giving direction to the user for a particular session. For example, user listens the first direction as *Right-up-diagonal* so at the first index (numbered as 1) of *Listensound* 3 will be stored.

Suppose prefix of user chosen password is *AF* and the first and the second both the challenge direction is *UP*. Now for the first character *A* the value of *K* (used in Algorithm 2) will be 0 and first index of array Listensound will contain value 2. So correct response by user will be (0 + 2) mod 10 = 2 against the first challenge *UP*. For the second character of user password *F* the value of *K* will be 5 and second index of array Listensound will contain value 2. So user will enter a value (5 + 2) mod 10 = 7 against the second challenge *UP* as per the Algorithm 2.

After evaluating the user response, system will decide whether the user is legitimate to login or not by using following Algorithm 3. It should be noted that before executing the Algorithm 2 the content of array *X* was 0. The user will get authenticated if value of *Y* is 1 after execution of Algorithm 3. Y will contain value 1 if user enters all responses correctly during login.

Algorithm 3. User Authentication

```
Input: This algorithm will take array X as input after executing Algorithm 2.
Output: Decides whether user is allowed to login.
Initialize Y = 0
for i = 1 to 5 do
  if X[$i$] = 1 then
    $Y \leftarrow 1$
  else
    $Y \leftarrow 0$
    break
  end if
end for
if Y = 1 then
  Allow user to login
else
  Disallow the user
end if
```

4 Important Features of SLASS

When user will enter a particular digit (say 0) as a response, attacker will not be able to retrieve any information either about challenge or about user chosen password as that digit (in this case 0) is placed at every table (from character *Table A* to character *Table J* in Fig. 2) and in all possible direction. This makes our proposed methodology robust against shoulder surfing attack.

System implemented for use in public domain demands efficiency in usability along with security. In our proposed methodology, we find it efficient against

Table 7. Time taken during learning

Number of users	Lesson Periods
15	1
8	2
5	3
2	more than 3

Table 8. User Feedback

Number of users	Feedback
23	Easy to understand
6	Fairly Understandable
1	Do not know

attacks like *Shoulder Surfing* or *guessing* the password. Our evaluation of usability and feedback from users also appears satisfactory. 30 users participated in our experiment. Among these users, 12 were college students and the rest were randomly picked from non-CS background. First we give a broad overview about how the methodology works. The feedback we got from most of the users is that our methodology is easy to understand. It should be noted that we only give the users lesson about how to use the system. Our lesson does not include security analysis of our system. Each lesson period was about 9 mins.

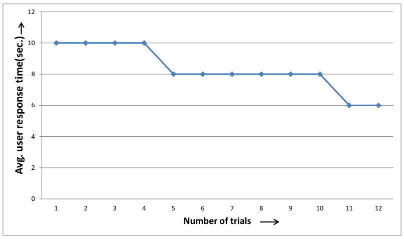


Fig. 4. Average Response Time

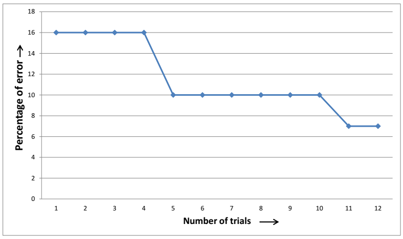


Fig. 5. Percentage of Error during Login

After a discussion with users, we give users about 30 mins to chose their password and for memorizing it. Then we asked the users to login with their password. We have performed our experiment in three phases. Number of trials given by all user under 5 times has been categorized under *first phase*. Number of trials between 5 to 10 times has been categorized under *second phase*. Number of trials greater than 10 times has been categorized under *third phase*. In our experiment average time of generating all the five challenges value was approximately 8 seconds. The *user response time* is the sum of the time taken to generate challenges and the response time by the user. After getting habituated with SLASS methodology the average user response time is 6 seconds and percentage of error during login is approximately 7% (shown in Fig. 4 and Fig. 5 respectively). Table 7 and Table 8 show an evaluation of understandability of our methodology in terms of learning.

5 Comparison with Existing Methodologies

Comparison of proposed methodology with classical password entry procedure shows considerable increase in login time but SLASS provides security against shoulder surfing attack. Probability of guessing the password in SLASS is $(1/10^5)$ as user remembers 5 characters instead of 4 (in case of classical password entry scheme). But if we implement SLASS using 4 character long password it will give same security as of classical password entry method in terms of guessing password. In this section we will compare SLASS with some of the others well known partially observable schemes [15], [13], [10] which give security against shoulder surfing attack.

In *mod 10 method* [15] user needs mathematical computational skill to login. So this might not be a simple method to adopt for non-math oriented people. As our methodology does not need any mathematical computation so it can be used by the non-math oriented people also. The minimum and maximum error during login using mod 10 method is above 20%. Whereas in SLASS the error reduced to 7%. Another variant of *mod 10 method* is *mod 10 table* [13]. The advantage of it is that it can be used by non-math oriented people also. But percentage of error during login is much higher than SLASS. Minimum error during login is is 15% . The password length in SLASS [10] and PIN length in SSSL are same i.e. 5 in both the cases. However, the probability of guessing password is more as SLASS uses 10 characters in its password whereas SSSL uses 9 digits in its PIN. The keyboard used in SLASS is similar with the keyboard used in classical password entry method. So SLASS has more user friendly interface than that of SSSL (keyboard structure in SSSL is given in Fig.1(b)).

6 Conclusion and Future Work

Security in public domain is a very important aspect. Normal password entry method is vulnerable to the shoulder surfing attack. To resist this kind of attack we have proposed a scheme which is robust against such attack. The proposed scheme is based on a partially observable attack model. As the complexity of SLASS is very less so the user response time is also very less. Our analysis also shows that the technique is easy to adopt. As our methodology does not require any mathematical computation from user's end so it can also easily be used by non-math oriented people. In the proposed scheme, user can choose a password from the set of 10 characters. But if all the 26 characters can be allowed that the user will get more freedom to choose his own password. In future we will try to achieve this objective.

Acknowledgement. This work is partially supported by a research grant from the Science and Engineering Research Board (SERB), Government of India, under sanction letter no. SB/FTP/ETA-226/2012.

References

1. Backes, M., Drmuth, M., Unruh, D.: Compromising reflections -or- how to read lcd monitors around the corner. In: Proceedings of the IEEE Symposium on Security and Privacy (SSP), Oakland, CA (May 2008)

2. Biddle, R., Chiasson, S., van Oorschot, P.: Graphical passwords: Learning from the first generation. technical report tr-09-09, school of computer science, carleton university (2009)
3. Blonder, G.E.: Graphical passwords. Lucent Technologies, Inc., Murray Hill, NJ, U. S. patent, ed. United States (June 1996)
4. Blum, L., Blum, M., Shub, M.: A simple unpredictable pseudo-random number generator. SIAM Journal on Computing 15(2), 364–383 (1986)
5. Herley, C., van Oorschot, P.C., Patrick, A.S.: Passwords: If we're so smart, why are we still using them? In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 230–237. Springer, Heidelberg (2009)
6. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and Other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
7. Li, Z., Sun, Q., Lian, Y., Giusto, D.D.: An association-based graphical password design resistant to shoulder surfing attack. In: IEEE International Conference on Multimedia and Expo. (ICME) (2005)
8. Mahansaria, D., Shyam, S., Samuel, A., Teja, R.: A fast and secure software solution [ss7.0] that counters shoulder surfing attack. In: 13th IASTED International Conference Software Engineering and Applications, pp. 190–195 (2009)
9. Paivio, A.: Mind and its evaluation: A dual coding theoretical approach (2006)
10. Perkovic, T., Cagali, M., Rakic, N.: SSSL: Shoulder surfing safe login. In: Software Telecommunications and Computer Networks, pp. 270–275 (2009)
11. Roth, V., Ritcher, K., Freidinger, R.: A pin-entry method resilient against shoulder surfing. In: ACM Conf. Comput. Commun. Security, pp. 236–245 (2004)
12. Tao, H., Adams, C.: Pass-Go:A proposal to improve the usability of graphical passwords. International Journal of Network Security 7(2), 273–292 (2008)
13. Perković, T., Čagalj, M., Saxena, N.: Shoulder-surfing safe login in a partially observable attacker model. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 351–358. Springer, Heidelberg (2010)
14. Wiedenbeck, S., Waters, J., Birget, J., Brodskiy, A., Memon, N.: Passpoints: Design and longitudinal evaluation of a graphical password system. Special Issue on HCI Research in Privacy and Security, International Journal of Human-Computer Studies (2005) (in press)
15. Wilfong, G.: Method and appartus for secure pin entry. US Patent No. 5,940,511. Lucent Technologies, Inc., Murray Hill, NJ, U. S. Patent, Ed. United States (1997)
16. Zhao, H., Li, X.: S3PAS: A scalable shoulder-surfing resistant textual-graphical password authentication scheme. In: 21st International Conference on Advanced Information Networking and Applications Workshops, pp. 467–472 (2007)
17. Zhou, Y., Feng, D.: Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing (2005)

Differential Execution Analysis for Obfuscation Reduction

R. Reno Robert

TIFAC CORE in Cyber Security, Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India
renorobert@gmail.com

Abstract. Code obfuscation is done to protect the logic of a program from easy analysis. A security analyst needs to spend considerable amount of time trying to de-obfuscate an executable by static analysis. The paper proposes the use of differential execution analysis to reduce obfuscation and constraint identification. Differential execution analysis filters critical instructions to analyze, from rest of instructions by comparing execution of program under different inputs using Dynamic Binary Instrumentation. After analysis, a reduced graph is generated out of dynamic execution trace showing reduced set of instructions as blocks separated by constraints placed on inputs.

Keywords: Obfuscation, Differential Execution, Dynamic Binary Instrumentation, Constraints, Reduced Graph.

1 Introduction

Obfuscation techniques are widely used to thwart reversing process. Its been used by software developers to protect critical logic of applications and also by malware writers to slow down a reverse engineer from analyzing the executable. Common obfuscation techniques include dead-code insertion, junk conditional and unconditional jumps, instruction substitution, constant hiding etc [7]. Some of these techniques are aimed at defeating disassembler features [8][9][10].

Differential execution analysis is used to reduce obfuscation by filtering out critical instructions from the dynamic execution trace. Reduced set of instructions will hugely assist a security analyst in finding the sections one has to concentrate on, during static analysis process. The obfuscated executable is monitored under different external input and each time, the trace of execution is generated with PIN Dynamic Binary Instrumentation(DBI) Framework [5][6]. Further data analysis is carried out of the dynamically generated traces to filter out instruction of interest. Unlike taint analysis [2][3][4] which identifies instructions acting on user data by keeping track of user data with taint propogation algorithm, differential execution analysis identifies user data by comparing executions of a program. Finally a graph is created out of dynamic execution trace with reduced set of critical instructions.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 358–369, 2014.

Proposed analysis technique is evaluted using an obfuscated sample program. Filtered set of instructions was found to be much less than the total instructions within the executable. This subset of instructions is easy to analyze than going through the entire disassembly.

2 Related Work

Differential Slicing [1] identifies difference in execution of same program under different input or environment. It outputs casual difference graph, showing the output difference due to input difference and casual path difference that lead from input difference to output difference. This assists in vulnerability analysis, as a security analyst can easily identify reason for crash.Aligned and unaligned regions are found from execution traces. Unaligned regions are difference in control flow. Noted difference in flow and values between two executions are used to generate casual difference graph, showing difference in value leading to flow difference.

Automated techniques for identifying cryptographic algorithms in executables is detailed in [11]. The authors perform fine-grained binary instrumentation using PIN DBI framework to generate trace of program execution including the instruction pointer, instruction disassembly, registers invloved, data handled, size and address of read or write operation. The generated trace is processed to identify the basic blocks and loops, then generate the control flow graph. Finally several heuristics are used to identify cryptographic code blocks present in the executable from dynamically generated trace of execution.

Various applications of differential static analysis is discussed in the paper [12]. One of the applications mentioned in the paper is debugging. Differential debugging could be used to narrow down failure inducing functions in a program by testing the program with different inputs, inputs which causes failures and inputs which do not.

In this paper, the use of differential execution analysis is studied to reduce obfuscation in executables. Fine-grained dynamic binary instrumentation is performed on the obfuscated sample under different inputs. Difference identification is carried out on the dynamically genrated traces to filter interesting code blocks from the obfuscated sample. This automated filtering of instructions overcomes the problem of static de-obfuscation of executable, which is a time consuming process.

3 Common Obfuscation Techniques

3.1 Junk Code Insertion

Junk code insertion adds instructions that does not change the meaning of code. Below is couple of instruction sequence that has no effect on the program

```
mov [rbp+var_80],r15
mov r15,[rbp+var_80]
mov [rbp+var_80],r15
mov r15,[rbp+var_80]

push r8
push r9
add rsp,0
pop r9
pop r8
```

The above sequence of data movement instructions does not change the meaning of program. If such instructions holds same value across multiple executions, it can be eliminated by differential execution analysis.

3.2 Unconditional Jumps

Inserting unconditional jumps in program makes linear disassembly technique unusable. By inserting jump statements, basic blocks can be broken down, making static analysis hard to perform. Trace generated during differential execution analysis removes unconditional jumps from the program, since execution path is deterministic

3.3 Conditional Jumps

Conditional jumps can change the control flow of the program based on constraints. Also during static analysis, the program logic has to be understood to decide which flow path is taken during execution. Conditional jumps can be inserted in a way that always a single path is taken

```
xor rax,rax
jz offset
```

3.4 Constant Hiding

Instead of directly hard coding constants into the program, constants can be replaced with expressions. Dynamic trace of execution reveals hidden constants in program and corresponding expression used to build constants can be removed by comparing executions.

4 Method Description

Dynamic analysis has the advantage that exact state of a process is known at a particular instance. Also only one execution path is analyzed based on the decision taken by the program. The main idea behind differential execution analysis is that, the data held by registers and memory varies when the program is run under different inputs. Not all instructions act on user data, the instructions that does not act on user data holds same value during each execution. Execution trace of program under analysis is obtained by supplying two different inputs. The generated execution trace holds the Instruction Pointer(IP), the instruction itself and the values held by its registers and memory.

Let Tx and Ty be the two traces obtained. Then traces Tx and Ty are compared till a difference in control flow is noticed or till the end of execution trace, when there is no difference in control flow. The output of this comparison is set of instructions that hold different values in both execution along with their values. A difference is control flow between two executions of same program occurs only when a decision is taken based on supplied input. For example, consider an *if* statement where a difference in control flow occurs with a Jcc instruction. These decision are made after a CMP or TEST instruction, which is considered as a constraint to input.

Finally a graph is generated from the dynamic execution trace with only reduced set of instructions. Instructions are split into blocks based on *constraint checking* instructions to highlight the constraint. The obfuscation reducer has three process - Trace Generation, Difference Identification and Graph creation. Architectural representation is shown in Fig. 1

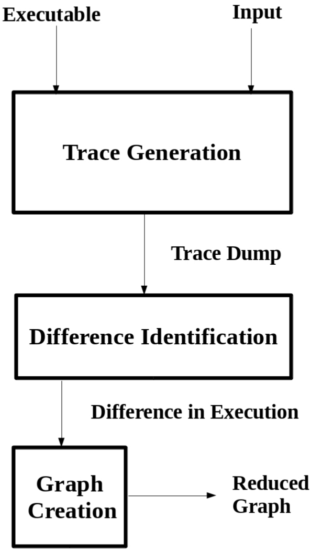


Fig. 1. Architecture

4.1 Trace Generation

Trace generation is implemented with Intel PIN Dynamic Binary Instrumentation framework. Every instruction executed by the program is logged for later analysis during difference identification phase. Format of trace is shown below

[Instruction Pointer] : [Instruction]
[Instruction Pointer] : [Data Held by Registers]

The point of start of trace can be entry point of executable or user defined range of address (Instruction Pointer). Since Instruction Pointer is used to identify the control flow difference in program, point of start must be the same for trace Tx and Ty. Different point of start will cause difference identification algorithm to fail straightaway.

PIN provides functions to classify instructions based on categories. An analyst can specify the categories of interest during trace generation phase. Some common categories of instructions that manipulate data are Binary, Logical, Rotate, Shift etc. Majority of instructions deal with data movements between memory or registers. Thus multiple levels of granularity can be achieved by specifying instruction category. Some instructions though deal with manipulating registers may not be of great interest for an analyst. For example, *sub rsp, immediate* is a common instruction used in prologue of function calls, to setup stack. Another example is *xor reg,reg* which is used for setting reg value to 0. These instructions are identified and removed during trace generation phase.

Address Space Layout Randomization(ASLR) is disabled during the trace generation process. Thus memory address of process is kept same during both the execution of program.

4.2 Difference Identification

The difference identification algorithm takes traces generated from the PIN instrumentation tracer and outputs only instructions that hold different values in both execution traces.

The algorithm iterates through each line in both the trace files Tx and Ty, simultaneously. Tx[I] and Ty[I] represents the Instruction Pointer(IP). When IP matches in both traces, the data held by its instruction operands, Tx[D] and Ty[D] are compared. The IP, instruction and data values of operands are added to trace Tz when a difference in data is found, else the instruction is skipped.

A difference in IP occurs due to control flow difference in execution. This difference in control flow occurs as some constraint is passed in one execution and failed in other ie. a conditional jump is taken, based on supplied input. Unconditional jumps though results in transfer of control flow, will end up in same IP in both execution. The algorithm stops when a difference in IP is noticed, no further comparison is made as different instruction deal with different register or memory operands and therefore different data. Now output trace Tz is generated, with only instructions and data that differs in trace Tx and Ty.

The trace will assist an analyst in finding the critical instructions and there is only lesser set of instructions to analyze. Since the trace Tz also includes data handled by registers, user inputs and constants used by program can be identified with less effort. This helps an analyst in setting breakpoints in debugger environment. To further simply analysis, a graph is generated from the filtered dynamic execution trace Tz which is detailed in next section.

```
Data: Traces Tx and Ty
Result: Filtered instruction trace Tz
I := 0
D := 1
while $Tx[I] == Ty[I]$ and not $EOF$ do
    if $Tx[D] \neq Ty[D]$ then
        filter(Tz, Tx[I], Tx[D])
        filter(Tz, Ty[I], Ty[D])
    end
    I := I + 2
    D := D + 2
end
```

Algorithm 1. Difference Identification Algorithm

4.3 Reduced Graph

Final stage of obfuscation reducer is the graph generation phase. Trace Tz obtained from Difference Identification phase is used for generating reduced graph. Reduced graph holds only the instructions that act upon external input.

Instructions in Tz are grouped into nodes based on constraints. Constraints are denoted with separate nodes to make them easy to identify. Each instruction in trace Tz is checked if its a constraint checking instruction, if not, NodeData is updated using UpdateNode(). UpdateNode() appends the current instruction to NodeData.

```
Index := GetNodeIndex()
AddNode(Block, NodeData, Index)
AddEdge(Block[Index-1], Block[Index])
NodeData := NULL
```

BuildGraph() function sets a unique index value for NodeData, adds NodeData as a node in reduced graph, creates edge to its previous node using index value. When a constraint checking instruction is identified, check if NodeData is empty. If NodeData is empty, update NodeData, then call BuildGraph() to add node and edges. If NodeData is not empty, create new node with existing data by calling BuildGraph(), then update NodeData and add the constraint as new node in graph. Such a graph generated out of dynamic execution trace will have only one execution path, since the conditional statements are already resolved. Loops and backward jump is represented in graph to help an analyst track execution flow.

```
Data: Trace Tz
Result: Graph with filtered instructions
I := 0, Index := 0, NodeData := NULL
Block := Dictionary(Index as Key, Node as Value)

while not EOF do
    if $Tz[I] \neq CMP$ then
        UpdateNode(Tz[I], NodeData)
    end
    else
        if $Node \neq NULL$ then
            BuildGraph()
            UpdateNode(Tz[I], NodeData)
            BuildGraph()
        end
        else
            UpdateNode(Tz[I], NodeData)
            BuildGraph()
        end
    end
    I := I + 2
end
```

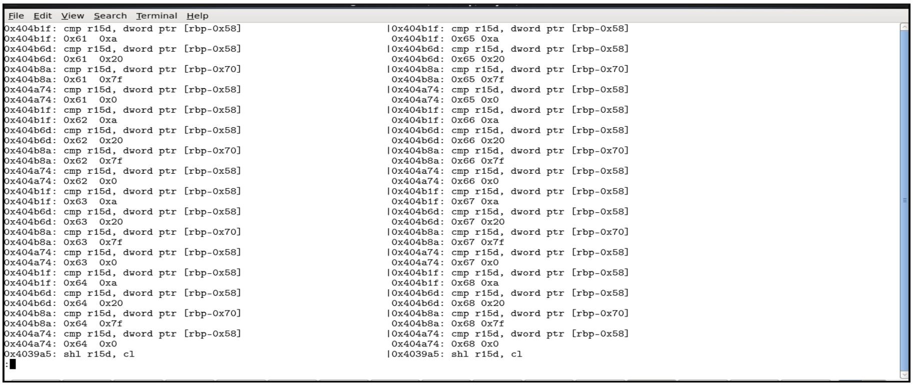
Algorithm 2. Graph Generation Algorithm

5 Analysis

Results of differential execution analysis is examined by testing it against a Linux ELF 64-bit obfuscated sample - *cone*, used in Plaid CTF 2013. Hopper graph view of the executable is shown in Fig.4 and Fig.5.

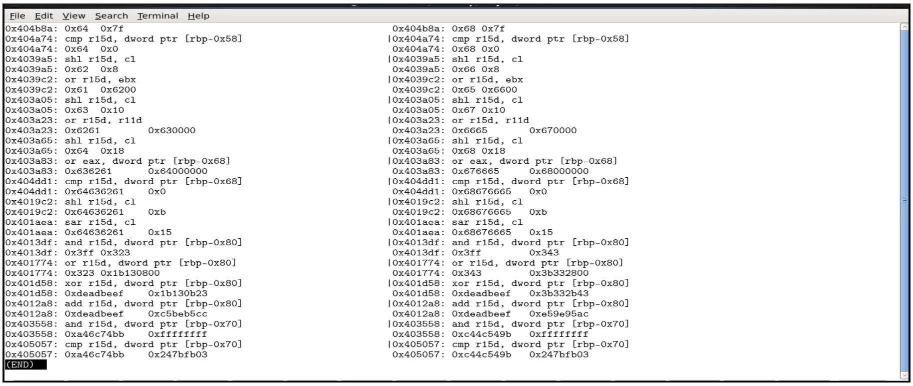
The executable under analysis had only few instructions of interest, but hidden inside lots of bogus instructions. It checks the user supplied value against an underlying algorithm for valid key. First, the trace of execution is obtained by supplying invalid keys *abcd* and *efgh* as inputs. Then differential execution analysis is carried out on the generated trace.

Trace was generated with data movement instructions removed. Fig.2 shows series of CMP instructions, comparing user input against some constant values. The repetitive sequence of comparison is carried out in a loop, this can be noticed by the repeating Instruction Pointer value. Fig.3 shows series of operations performed on user input, which is the underlying algorithm. The values that remain same across both the execution are constants used by program and does not depend on user input. CMP instructions can be seen constraints placed on user input. An analyst can figure out constraints that causes difference in control flow. By identifying input that passes or fails a constraint, differential execution analysis can be carried out further in desired execution path by supplying suitable inputs. The output of difference identification phase is shown in Fig.2 and Fig.3.



```
0x404b1f: cmp r15d, dword ptr [rbp-0x58]                        |0x404b1f: cmp r15d, dword ptr [rbp-0x58]
0x404b1f: 0x61  0xa                                              0x404b1f: 0x65 0xa
0x404b6d: cmp r15d, dword ptr [rbp-0x58]                        |0x404b6d: cmp r15d, dword ptr [rbp-0x58]
0x404b6d: 0x61  0x20                                             0x404b6d: 0x65 0x20
0x404b8a: cmp r15d, dword ptr [rbp-0x70]                        |0x404b8a: cmp r15d, dword ptr [rbp-0x70]
0x404b8a: 0x61  0x7f                                             0x404b8a: 0x65 0x7f
0x404a74: cmp r15d, dword ptr [rbp-0x58]                        |0x404a74: cmp r15d, dword ptr [rbp-0x58]
0x404a74: 0x61  0x0                                              0x404a74: 0x65 0x0
0x404b1f: cmp r15d, dword ptr [rbp-0x58]                        |0x404b1f: cmp r15d, dword ptr [rbp-0x58]
0x404b1f: 0x62  0xa                                              0x404b1f: 0x66 0xa
0x404b6d: cmp r15d, dword ptr [rbp-0x58]                        |0x404b6d: cmp r15d, dword ptr [rbp-0x58]
0x404b6d: 0x62  0x20                                             0x404b6d: 0x66 0x20
0x404b8a: cmp r15d, dword ptr [rbp-0x70]                        |0x404b8a: cmp r15d, dword ptr [rbp-0x70]
0x404b8a: 0x62  0x7f                                             0x404b8a: 0x66 0x7f
0x404a74: cmp r15d, dword ptr [rbp-0x58]                        |0x404a74: cmp r15d, dword ptr [rbp-0x58]
0x404a74: 0x62  0x0                                              0x404a74: 0x66 0x0
0x404b1f: cmp r15d, dword ptr [rbp-0x58]                        |0x404b1f: cmp r15d, dword ptr [rbp-0x58]
0x404b1f: 0x63  0xa                                              0x404b1f: 0x67 0xa
0x404b6d: cmp r15d, dword ptr [rbp-0x58]                        |0x404b6d: cmp r15d, dword ptr [rbp-0x58]
0x404b6d: 0x63  0x20                                             0x404b6d: 0x67 0x20
0x404b8a: cmp r15d, dword ptr [rbp-0x70]                        |0x404b8a: cmp r15d, dword ptr [rbp-0x70]
0x404b8a: 0x63  0x7f                                             0x404b8a: 0x67 0x7f
0x404a74: cmp r15d, dword ptr [rbp-0x58]                        |0x404a74: cmp r15d, dword ptr [rbp-0x58]
0x404a74: 0x63  0x0                                              0x404a74: 0x67 0x0
0x404b1f: cmp r15d, dword ptr [rbp-0x58]                        |0x404b1f: cmp r15d, dword ptr [rbp-0x58]
0x404b1f: 0x64  0xa                                              0x404b1f: 0x68 0xa
0x404b6d: cmp r15d, dword ptr [rbp-0x58]                        |0x404b6d: cmp r15d, dword ptr [rbp-0x58]
0x404b6d: 0x64  0x20                                             0x404b6d: 0x68 0x20
0x404b8a: cmp r15d, dword ptr [rbp-0x70]                        |0x404b8a: cmp r15d, dword ptr [rbp-0x70]
0x404b8a: 0x64  0x7f                                             0x404b8a: 0x68 0x7f
0x404a74: cmp r15d, dword ptr [rbp-0x58]                        |0x404a74: cmp r15d, dword ptr [rbp-0x58]
0x404a74: 0x64  0x0                                              0x404a74: 0x68 0x0
0x4039a5: shl r15d, cl                                          |0x4039a5: shl r15d, cl
:
```

Fig. 2. Difference Identification using Dynamic Trace I



```
0x404b8a: 0x64  0x7f                                             0x404b8a: 0x68 0x7f
0x404a74: cmp r15d, dword ptr [rbp-0x58]                        |0x404a74: cmp r15d, dword ptr [rbp-0x58]
0x404a74: 0x64  0x0                                              0x404a74: 0x68 0x0
0x4039a5: shl r15d, cl                                          |0x4039a5: shl r15d, cl
0x4039a5: 0x62  0x8                                              0x4039a5: 0x66 0x8
0x4039c2: or r15d, ebx                                          |0x4039c2: or r15d, ebx
0x4039c2: 0x61  0x6200                                           0x4039c2: 0x65 0x6600
0x403a05: shl r15d, cl                                          |0x403a05: shl r15d, cl
0x403a05: 0x63  0x10                                             0x403a05: 0x67 0x10
0x403a23: or r15d, r11d                                         |0x403a23: or r15d, r11d
0x403a23: 0x6261        0x630000                                 0x403a23: 0x6665       0x670000
0x403a65: shl r15d, cl                                          |0x403a65: shl r15d, cl
0x403a65: 0x64  0x18                                             0x403a65: 0x68 0x18
0x403a83: or eax, dword ptr [rbp-0x68]                          |0x403a83: or eax, dword ptr [rbp-0x68]
0x403a83: 0x636261      0x64000000                               0x403a83: 0x676665     0x68000000
0x404dd1: cmp r15d, dword ptr [rbp-0x68]                        |0x404dd1: cmp r15d, dword ptr [rbp-0x68]
0x404dd1: 0x64636261    0x0                                      0x404dd1: 0x68676665   0x0
0x4019c2: shl r15d, cl                                          |0x4019c2: shl r15d, cl
0x4019c2: 0x64636261    0xb                                      0x4019c2: 0x68676665   0xb
0x401aea: sar r15d, cl                                          |0x401aea: sar r15d, cl
0x401aea: 0x64636261    0x15                                     0x401aea: 0x68676665   0x15
0x4013df: and r15d, dword ptr [rbp-0x80]                        |0x4013df: and r15d, dword ptr [rbp-0x80]
0x4013df: 0x3ff 0x323                                            0x4013df: 0x3ff        0x343
0x401774: or r15d, dword ptr [rbp-0x80]                         |0x401774: or r15d, dword ptr [rbp-0x80]
0x401774: 0x323 0x1b130800                                       0x401774: 0x343        0x3b332800
0x401d58: xor r15d, dword ptr [rbp-0x80]                        |0x401d58: xor r15d, dword ptr [rbp-0x80]
0x401d58: 0xdeadbeef    0x1b130b23                               0x401d58: 0xdeadbeef   0x3b332b43
0x4012a8: add r15d, dword ptr [rbp-0x80]                        |0x4012a8: add r15d, dword ptr [rbp-0x80]
0x4012a8: 0xdeadbeef    0xc5beb5cc                               0x4012a8: 0xdeadbeef   0xe59e95ac
0x403558: and r15d, dword ptr [rbp-0x70]                        |0x403558: and r15d, dword ptr [rbp-0x70]
0x403558: 0xa46c74bb    0xffffffff                               0x403558: 0xc44c549b   0xffffffff
0x405057: cmp r15d, dword ptr [rbp-0x70]                        |0x405057: cmp r15d, dword ptr [rbp-0x70]
0x405057: 0xa46c74bb    0x247bfb03                               0x405057: 0xc44c549b   0x247bfb03
(END)
```

Fig. 3. Difference Identification using Dynamic Trace II

The total number of instructions logged during trace generation phase was 2566, out of which 19 instructions where filtered. Results are tabulated in Table 1.

As the final stage of obfuscation reducer, a graph is generated from the output of difference identification phase. Reduced graph for executable under analysis is shown in Fig.6.

Table 1. Instruction Reduction

Instructions Executed	Instructions Filtered	Percentage of Reduction
2566	19	99.26

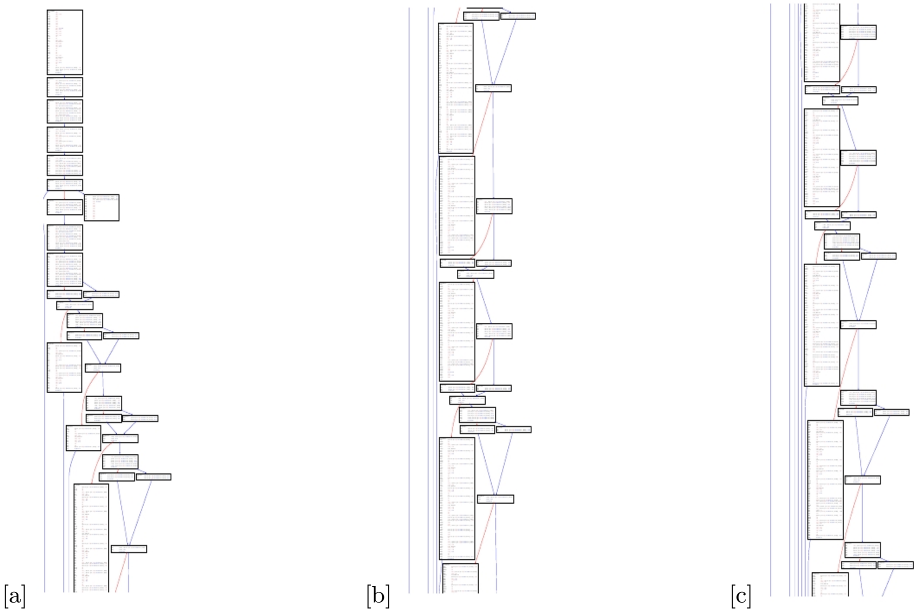


Fig. 4. Sections of Control Flow Graph (generated by hopper disassembler) of obfuscated executable under analysis

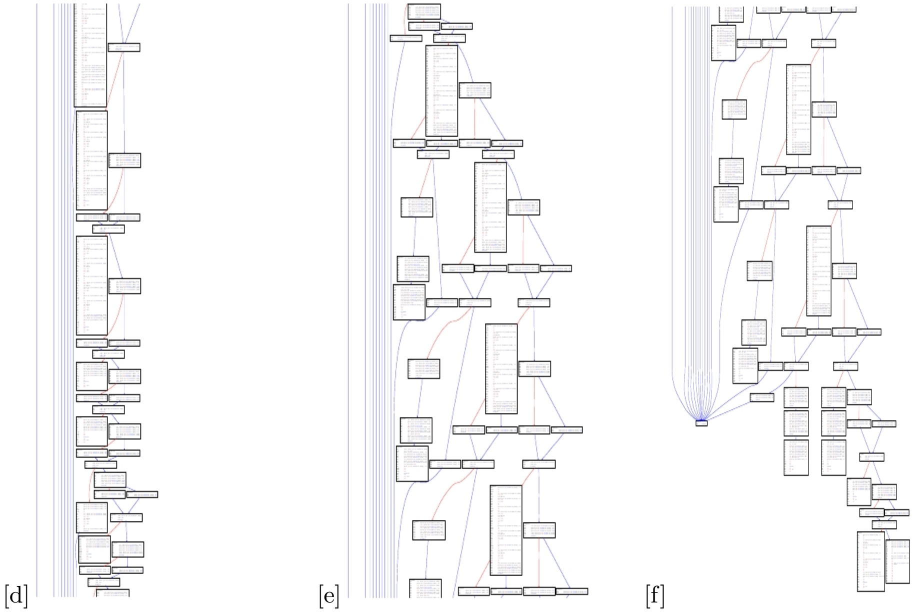
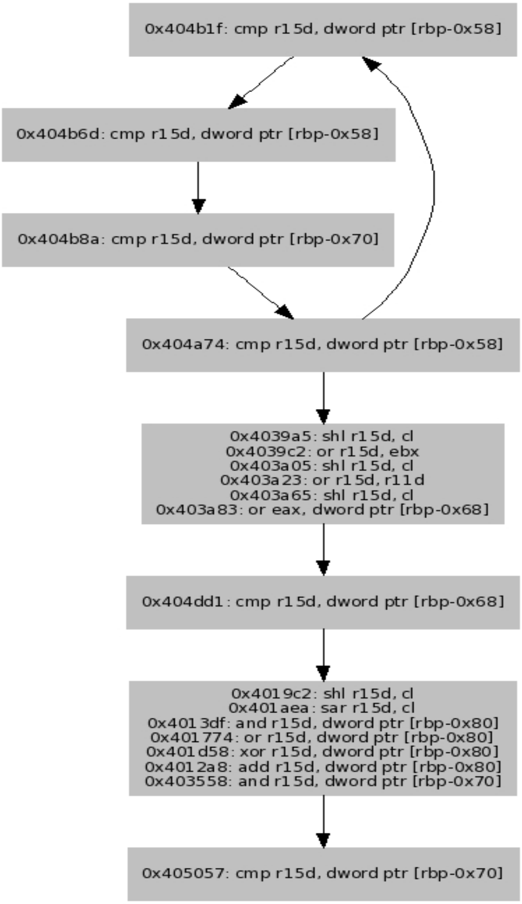


Fig. 5. Sections of Control Flow Graph (generated by hopper disassembler) of obfuscated executable under analysis



```
0x404b1f: cmp r15d, dword ptr [rbp-0x58]
0x404b6d: cmp r15d, dword ptr [rbp-0x58]
0x404b8a: cmp r15d, dword ptr [rbp-0x70]
0x404a74: cmp r15d, dword ptr [rbp-0x58]
0x4039a5: shl r15d, cl
0x4039c2: or r15d, ebx
0x403a05: shl r15d, cl
0x403a23: or r15d, r11d
0x403a65: shl r15d, cl
0x403a83: or eax, dword ptr [rbp-0x68]
0x404dd1: cmp r15d, dword ptr [rbp-0x68]
0x4019c2: shl r15d, cl
0x401aea: sar r15d, cl
0x4013df: and r15d, dword ptr [rbp-0x80]
0x401774: or r15d, dword ptr [rbp-0x80]
0x401d58: xor r15d, dword ptr [rbp-0x80]
0x4012a8: add r15d, dword ptr [rbp-0x80]
0x403558: and r15d, dword ptr [rbp-0x70]
0x405057: cmp r15d, dword ptr [rbp-0x70]
```

Fig. 6. Graph generated out of dynamic execution trace with reduced instruction set and loops identified

Instructions are displayed along with its address. Constraints are displayed as separate nodes and rest of the instructions are grouped into single node till a constraint checking instruction is found as per the algorithm explained in section 4. Now the analyst can work with information obtained from difference identification phase and reduced graph to simplify the task of de-obfuscation and identify interesting parts of the executable.

6 Conclusion and Future Work

The paper explains differential execution analysis technique to reduce obfuscation in executable. Execution of a program is traced under different supplied inputs and generated traces are compared to find instructions acting on user data. Bogus instructions are removed and analyst is provided with reduced set of instructions which is a subset of entire set instructions in the program.

Differential execution analysis is found to be effective against tested sample but further research could be carried out to extend the technique to analyze obfuscated malware samples, where trace would be large. Also, use of theorem provers to solve constraints would further ease the job of an analyst and provide a better analysis technique.

Acknowledgements. I would like to thank Dr. Gireesh Kumar for help pursuing my research interest and Dhanesh Kizhakkinan for various discussions and help with reverse code engineering during the course of this work.

References

1. Johnson, N.M., Caballero, J., Chen, K.Z., McCamant, S., Poosankam, P., Reynaud, D., Song, D.: Differential Slicing: Identifying Causal Execution Differences for Security Applications. In: Proceedings of the 2011 IEEE Symposium on Security and Privacy, pp. 347–362 (2011)
2. Kemerlis, V.P., Portokalidis, G., Jee, K., Keromytis, A.D.: libdft: Practical Dynamic Data Flow Tracking for Commodity Systems. In: Proceedings of the 8th ACM SIGPLAN/SIGOPS Conference on Virtual Execution Environments, pp. 121–132 (2012)
3. Clause, J., Li, W., Orso, A.: Dytan: A Generic Dynamic Taint Analysis Framework. In: Proceedings of the 2007 International Symposium on Software Testing and Analysis, pp. 196–206 (2007)
4. Saxena, P., Sekar, R., Puranik, V.: Efficient Fine-Grained Binary Instrumentation with Applications to Taint-Tracking. In: Proceedings of the 6th Annual IEEE/ACM International Symposium on Code Generation and Optimization, pp. 74–83 (2008)
5. Luk, C.-K., Cohn, R., Muth, R., Patil, H., Klauser, A., Wallace, S., Reddi, V.J., Hazelwood, K., Lowney, G.: Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation. In: Proceedings of the 2005 ACM SIGPLAN Conference on Programming Language Design and Implementation, pp. 190–200 (2005)
6. PIN, Dynamic Binary Instrumentation Framework by Intel, `http://software.intel.com/en-us/articles/pin-a-dynamic-binary-instrumentation-tool`
7. You, I., Yim, K.: Malware Obfuscation Techniques: A Brief Survey. In: Proceedings of the 2010 IEEE Conference on Broadband, Wireless Computing, Communication and Applications, pp. 297–300 (2010)
8. Linn, C., Debray, S.: Obfuscation of Executable Code to Improve Resistance to Static Disassembly. In: Proceedings of 10th ACM Conference on Computer and Communications Security, pp. 290–299 (2003)
9. Kruegel, C., Robertson, W., Valeur, F., Vigna, G.: Static Disassembly of Obfuscated Binaries. In: Proceedings of the 13th Conference on USENIX Security Symposium, pp. 255–270 (2004)

10. Udupa, S.K., Debray, S.K., Madou, M.: Deobfuscation: Reverse Engineering Obfuscated Code. In: Proceedings of the 12th Working Conference on Reverse Engineering, pp. 45–54 (2005)
11. Gröbert, F., Willems, C., Holz, T.: Automated identification of cryptographic primitives in binary programs. In: Sommer, R., Balzarotti, D., Maier, G. (eds.) RAID 2011. LNCS, vol. 6961, pp. 41–60. Springer, Heidelberg (2011)
12. Lahiri, S.K., Vaswani, K., Hoare, C.A.R.: Differential Static Analysis: Opportunities, Applications and Challenges. In: Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, pp. 201–204 (2010)

Emergency Aware, Non-invasive, Personalized Access Control Framework for IMDs

Monika Darji[1] and Bhushan H. Trivedi[2]

[1] LJ Institute of Computer Application
Ahmedabad, India
monikadarji79@gmail.com
[2] GLS Institute of Computer Technology
Ahmedabad, India
bhtrivedi@yahoo.com

Abstract. Implantable Medical Devices have helped patients suffering from chronic diseases by providing continuous diagnosis, treatment and remote monitoring without hospitalization and at a less expense with increased flexibility. Incorporation of wireless bidirectional communication has introduced vulnerabilities like unauthorized wireless access which might get realized as a security attack and endanger patient privacy and safety. Traditional security and privacy techniques cannot be directly applied to these devices because of their miniaturized size which leads to power, computational and storage constraint. Moreover their positioning inside the human body makes battery replacement possible only through surgery. Security and privacy technique for these devices must balance security and safety and should also be acceptable and usable. Moreover it should not reduce the clinical effectiveness of the device. Security researchers have proposed ways of providing security but have kept the property of fail openness in order to make IMD accessible during emergencies. Fail openness is defined as a property of Implantable Medical Device due to which during emergency condition access is granted bypassing all security techniques. We argue that the patient is all the more vulnerable during an emergency situation and complete removal of security may be dangerous for the safety of the patient.We propose a solution to provide fine grained Access Control which also takes emergency condition into notice. The security needs for IMD communication requires dynamic and flexible policy enforcement. While providing strong Access Control during normal situation, our solution accommodates emergency access to the data in a life-threatening situation. We propose personalized Emergency Aware role based Access Control (EAAC) framework. This framework can work in conjunction with Authentication and Encryption to provide a strong security solution as compared to other solutions. In fact we believe that the possibility of an attacker inducing false alarms to introduce fake emergency situation and take control of the IMD is likely to increase and the solution that we propose here may be more useful in such cases.Our paper highlight security challenges when fail open access is given and provide a solution using EAAC framework.

Keywords: IMDs, security, authentication, access control.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 370–381, 2014.

1 Introduction

Implantable Medical Devices, as the name suggests, are implanted in the patient's body for therapeutic use and while in the body, performs the task of sensing, actuation, communication with outside readers/programmers or with other IMDs. IMDs are enabled with wireless capabilities to communicate using the MICS band with frequency range of 402 to 405 MHz and maximum bandwidth 300 kHz [25]. Patients with IMD are monitored automatically, continuously and remotely using wireless communication in quite a few systems today. There were around 245,000 insulin pump users in 2005,and which is expected to grow at a compound rate of 9% from 2009 to 2016 [1]. For example a Cardioverter Defibrillators (ICD) is capable of sensing heartbeat and administering an electrical shock to restore a normal heart rhythm, medical practitioner can use device programmer to extract data or modify settings wirelessly [10]. The use of wireless communication to monitor and interact with IMD makes them vulnerable to exploitation by hackers and tech-criminals. Authentication and Access control is therefore an essential component of IMDs security mechanism to prevent unauthorized access to wireless telemetry and illegitimate issue of control commands to IMD. The term authentication means verifying the identity of a person is the same that it claims to be and the term Access Control is used to restrict the actions a legitimate entity can perform on a given resource object [12]. We categorize a patient implanted with one or more IMDs into two states: Normal and Emergency [8]. In Normal State patient's IMD can be accessed by following normal security protocol, in Emergency State when the patient is unconscious and therefore is not in a position to manage his IMD security [10][5] and to access IMD, authorized staff is unavailable. An unauthorized person can gain control of an IMD's operation or even disable its therapeutic services, for example he may compromise ICD to deliver disruptive electrical shocks to a heart [10]. Researchers have also demonstrated compromise over an insulin pump, including the ability to stop insulin delivery or inject excessive doses [3, 4]. Due to resource constraints on IMD, we argue that instead of placing Access Control logic on IMD we can place it on a handheld device like a cell phone or PDA which a patient can carry. Access Control in Normal state can follow standard pre-defined Access Control policies such as Role-Based Access Control defined in (or used in) [9]. However when the patient is in Emergency State, service requirements may be different. The standard Access Control policy may prevent an unauthorized practitioner to access the patient's IMD for administering emergency treatment. The Access Control mechanism should be able to observe the environmental conditions to detect changes i.e. emergency conditions which cause the patient to enter into an Emergency State. In the Emergency State, Access Control policies need to be modified dynamically to allow quick access and immediate treatment. Access Control policy changes in response to the Emergency State should be temporary and regular Access Control policy should restore once the emergency is over. Therefore the model for Access Control should not only prevent unauthorized access to the IMD but also provide facilities for responding to the effects of critical events which may lead to Emergency. Criticality Aware Access Control (CAAC) [18] is a framework that can be used to fulfill these needs. The goal of this paper is to leverage the framework provided in CAAC for specifying Access Control policies that control access to IMDs in both Normal and Emergency State and extend it towards more automated means of working while emergency. Another important issue is the placement of Access Control Service. As the framework requires continuous monitoring of context, authentication and application of Access Control policy for secure access to IMD telemetry, instead of placing the service on the IMD itself we propose to put it on an external proxy device. It will also beneficial when a patient has multiple IMDs installed as instead of every IMD managing its security, a centralized rechargeable proxy can easily may do so for all of them. We propose the use of a proxy-device like a PDA or cell phone which has an Internet access for performing Access Control functionality on behalf of IMD. By shifting security related processing tasks to the proxy-device, we can reduce IMD cost and energy consumption. In Normal state, Proxy Device performs Role Based Access Control and in Emergency State if regular medical aid is not available, Proxy device uses the Internet to get connected to its localized Virtual Space where it will give emergency access to an authorized medical practitioner by providing him with a temporary credential to access the Patient IMD for immediate treatment instead of failing open to give access to everybody. During emergency this security system will ensure Non-repudiation as well. A diagrammatic representation of the model is given below:

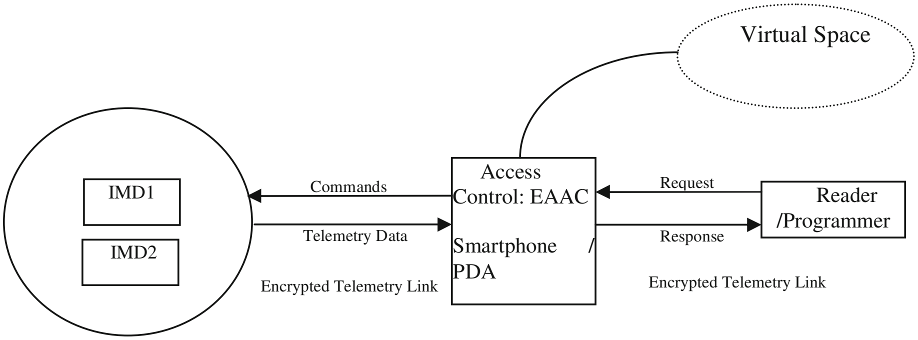


Fig. 1. Block Diagram for Emergency Aware Access Control using Proxy Device

2 Contribution and Organization of the Paper

Our contribution in this paper is to point out security vulnerabilities when IMDs communicate wirelessly, providing justifications on aggravated risk for patients when no security mechanism is used during emergency and finally extending the CAAC [18] Access Control model for IMDs to take into account the effect of emergency situation along with discussion about its placement on a PDA or cell phone which can be carried by the patient. The rest of the paper is organized as follows: In Section 3, we survey the Access Control techniques proposed in literature to prevent illegitimate access. In Section 4, we present a Threat Model for fail open Access Control and our assumptions. In Section 5, we present the Security Mechanism proposed to be installed on Proxy Device. In Section 6, we present the EAAC architectural framework and Section 7 concludes the paper.

3 Survey of Access Control Techniques for IMD Literature

Here we mention the Access Control techniques proposed in literature to prevent illegitimate access to IMD. In [6], Medical staff accesses the IMD using an access token which can be a USB stick or bracelet configured with a secret key, to establish a secure encrypted link to download data and send programming commands. Access is granted to anybody who holds the token. These access tokens need to be protected from theft and forgery, if lost or stolen or forgotten, it creates a safety problem by rendering the IMD inaccessible. In [7], Password is proposed to be tattooed as ultraviolet-ink micropigmentation, adjacent to the point of implantation as a UV-visible 2D barcode which are invisible under normal light. Devices that interact with IMD must be equipped with ultraviolet light emitting diode (UV LED) and have an input mechanism for key entry. This technique mentions the risk of infection for patients from micropigmentation and the risk that a tattoo could be rendered unreadable when needed. In the security solutions mentioned above, if the password is unavailable, IMD becomes inaccessible leading to safety issues, moreover there are no rules and policies defined for rendering access control. Also the password cannot be renewed.

The scheme mentioned in [8] uses proximity-based device pairing protocol based on ultrasonic distance bounding which allows access to only those devices which are in close proximity. This technique enables the IMDs to predefine an exact range for granting access and prevents the attacker from accessing the IMD from distance regardless of the type of transceiver he has. IMD can operate in two different modes, in normal mode, the reader needs to be in possession of a shared key to access the IMD and in the emergency mode reader just needs to be within certain security range. The drawbacks are that it requires and assumes ultrasonic distance bounding features to be implemented on IMD, ultrasonic distance bounding is vulnerable to RF wormhole attacks, and further this protocol is vulnerable to Denial of Service to actual reader. When IMD detects an emergency situation (stroke, heart failure, etc.), Access Control is deactivated all together, this aggravates the risk of security attacks.

In the technique described in [9] patients are made aware of any RF activity happening with the IMDs. Zero-power notification harvests induced RF energy to wirelessly power a piezo-element that audibly alerts the patient of security-sensitive events at no cost to the IMDs battery. Zero-power authentication uses symmetric cryptographic techniques to prevent unauthorized access; it aims to protect against adversaries who have custom equipment. The sensible key exchange involves vibration-based key distribution that a patient can sense through audible and tactile feedback. The audible alerts may not be very helpful in noisy environment. This technique does not address the problem of key management, renewal and revocation also key exchange may be susceptible to eavesdropping. Access Control is provided through patient notification as a side-effect of a cryptographic operation as a combination of audible and tactile feedback which may not be useful if the patient is unconscious.

In the system described in [10], a removable device called communication cloaker is used for defensive countermeasure that controls access to the IMD by making it invisible to all unauthorized queries. This device provides security when worn by the patient, makes IMD act on commands sent by only authorized medical staff, encrypts all communications to and from the IMD and checks them for authenticity and integrity. When the cloaker is removed, it provides fail open access to all external programmers. In this way emergency medical staff can access a patient's IMD even if the wristband is lost or destroyed in an accident. If a doctor needs to access the IMD, the patient has to remove his cloaker to allow communication. This approach provides a solution to the security/safety tension but provides no Access Control and security when Cloaker is removed. This scheme allows open access in emergency situations but simultaneously increases vulnerability as long as the device is openly accessible.

The system described in [11] delegates the security of an IMD to a personal base station called the shield. The shield act as a jammer-cum-receiver to jam the IMDs messages and unauthorized commands preventing others from decoding them while itself being able to decode them. Here also, to give fail open access the shield is removed during an emergency.

The secure telemetry link solution in [20] proposes to use a physical backdoor to verify that the reader is close to the IMD. When the reader wants to access the IMD, it sends an "activation message", over the wireless channel to the IMD that activates the backdoor circuitry. The reader then gets close to the IMD using a near-field magnetic sensor. , IMD sends the authentication key over the wireless link, using a very low transmission power, on detecting the reader's sensor. The reader uses the key to communicate over a secure channel. It is not secure against attacks that make use of special devices like a high-gain antenna to eavesdrop the key moreover no authentication is performed before sharing the secret key.

Other schemes [15, 23] also use short-range communication technologies (such as IR, Bluetooth, RFID, etc.) to guarantee proximity and are therefore not secure. These approaches are vulnerable to attacks that make use of high gain antennas and advanced transceivers to access IMDs from a greater distance.

From this survey we conclude that there is a strong need of emergency aware Access Control.

4 Threat Model for Fail Open Security

One obstacle with guaranteeing the security of an implantable medical device is that the list of authorized parties may change depending on the patient's circumstances, as discussed at length in [2].The literature shows most of the security solution proposed for IMDs encompasses the property of fail openness during emergencies so that a medical practitioner can have immediate access to the IMDs. Nothing is mentioned about when security solution will be put back in place. Also by bringing down the security to zero means introducing vulnerability as during such circumstance IMD will communicate without using any cryptographic mechanism making the communication vulnerable to eavesdropping. It will not perform any authentication therefore anyone is allowed to communicate with IMD increasing risks of Insider and Outsider attacks. As no Access Control is ascertained, people with the right kind of hardware can play havoc with the device. There is an equal possibility of attacks on integrity, replay attacks, denial of service attacks and non-repudiation cannot be guaranteed. This can turn out to be a big loophole in the entire framework of security in IMD and consequences can be severe. In fact we believe that the possibility of an attacker inducing false alarms to introduce fake emergency situation and take control of the IMD is likely to increase in future. The resort to fail open cannot be the solution for emergency any longer.

Assumption

As our concern is Access Control we assume a strong Authentication Mechanism in place to authenticate user in Normal State. We assume that Proxy device is able to communicate to the IMD using a secure encrypted channel and Proxy device is capable of using an Internet connection to access Virtual Space which keeps a record of medical practitioners in that location.

5 Security Mechanisms Proposed to Be Installed on Proxy Device

5.1 Authentication

A method of authenticating readers/programmers is a prerequisite for enforcing access control. When the IMD is operating normally in a non-emergency situation, authentication can be handled by requiring proof that the authenticating party is legitimately representing itself. This authentication is typically guaranteed by requiring the authenticating party to provide one or more of three factors: something known (like a password), something possessed (like a physical key), or something unique about the party (like fingerprint) [24]. The other techniques used are biometrics (fingerprints, iris scan, signature and voice recognition and digital techniques (e-tokens, RFID, key fobs) [17]. Authentication must be given using the principle of "least privilege", i.e. privilege should be withheld until its need is established by a specific request, e.g. even if a doctor(provider) logs in he should not be given privilege of stopping IMD as he may accidently do so. If and when the user makes a request that requires a higher trust level, the user is offered the opportunity to re-authenticate using any available method of higher reliability to gain a new trust credential with a higher trust level [17].Authentication Service must reliably identify the user and issue a credential [16]. This credential accompanies every request for access that a reader/programmer makes, and is used by the Access Control Service to identify the requester. The specific authentication mechanisms are not the focus of this paper; we can use any authentication mechanism that securely transmits the credential to the user, in such a manner that credentials cannot be tampered with, stolen, or forged. Given these properties, the Access Control system can rely on the credential to identify the requester.

5.2 Access Control

To provide secure communication, Access Control to IMD is crucial. A typical Access Control system consists of a subject, an object, permission, and credentials [18]. A subject is an entity (reader/programmer) that seeks access to a resource object (IMD). An object is actually a target protected by the Access Control system and here it is the IMD. Permission is an access right for a subject over an object that corresponds to a privilege that a subject owns over a certain object.

Access Control solutions based on close-range communication have the advantage of being simple and intuitive [8], [9] but do not provide any guarantees about the range of communication. Namely, an attacker with a strong enough transmitter and a high-gain antenna may still be able to communicate with the IMD even from far outside the intended range, solutions based on magnetic switches are also based on close-range communication; in addition they do not require any form of authentication to unlock access to the device and are thus inherently insecure [9].

An emergency-aware Access Control model that incorporates the contextual information in controlling access to sensitive IMDs is adaptive in response to dynamic changes in the patient contextual information. The IMD may be configured to provide diagnostic information to the proxy device if it detects a weak pulse, low blood glucose, or if the patient is unconscious. In these circumstances, the proxy may switch from Normal State to Emergency State.

Below are the Access Control mechanisms and a discussion on the suitability of their application in implementation of Access Control for IMD:

Traditional Rule Based Model

Discretionary Access Control (DAC) model uses Access Control Matrix (ACM) that defines the access rights of each subject over a set of resource object on the resource owner’s discretion [12]. It is a static Access Control solution where subjects and objects need to be pre-defined. As the Access Control decisions are immutable, additional constraints cannot be imposed easily. Another solution is called Mandatory Access Control (MAC) access rights are determined by a central authority by labeling each resource object with a sensitivity level and each subject with a clearance level [13]. Resource object is accessible only to a subject that possesses a valid clearance level. These models lack dynamism and flexibility and cannot be used to provide Access Control for IMD.

Role Based Access Control Model

In this solution access to a resource object is governed based on subject's role [14]. As access right is not assigned to subjects directly but to roles administration and modification is easy. In RBAC, subjects in the system are assigned roles when they join the system, and are allowed to access resources, within the system, based on the privileges associated with the roles. Given a set of roles and privileges RBAC involves two main activities: mapping subjects to roles and mapping the roles to different sets of privileges. RBAC constraints can be imposed on access request to prevent unauthorized access. It ensures security against leakage of access rights to unauthorized entities and also ensures integrity. This model is not able to incorporate dynamic assess control as mappings are static and cannot change depending on the context in which access request is made and current situation of the patient.[25] describes an authorization model based on semantic web technologies which represents the underlying information model using Common Information Model (CIM). For managing the authorization of resources correctly, authorization systems should implement its semantics so as to provide a match with semantics of underlying data and resources to be protected. [26] extends the RBAC model in order to support highly complex privacy-related policies, taking into account features like purposes and obligations.

Context Aware Access Control Model

CA-RBAC extends RBAC model to accommodate contextual information for controlling access to sensitive resource objects. This solution is flexible and dynamic, depends on combination of required credentials of user and context and state of the system. CA-RBAC model proposed in [15] considered the spatial, temporal and resource context in Access Control decision making. [16] Presents an infrastructure for context aware Access Control and authentication in smart spaces. A dynamic context aware Access Control scheme for distributed healthcare applications was presented in [17].

Criticality Aware Access Control Model (CAAC)

This framework automatically responds to occurrences of critical events and change Access Control policies accordingly [18]. It includes a new parameter called Criticality which measures the urgency of tackling the effects of a critical event. This model claims to meet Responsiveness, Correctness, Non-interference, Liveness and Non-repudiability. CAAC classifies system context information into two main categories: Critical and Non-Critical. Critical contexts indicate the setting or the occurrence of a critical event which requires immediate action. Noncritical contexts, on the other hand, are observed during normal system states and require no special action. CAAC provides the ability to automatically adjust Access Control policies in response to critical events. This may include notification, logging, Access Control relaxations/restriction, and more depending on application requirements. Our Access Control framework is based on CAAC.

6 Proposed EAAC Architectural Framework

We propose emergency aware Access Control framework on a proxy device like a PDA / cell phone which acts as a man-in-the-middle to grant access to one or more IMDs implanted on the Patient's body. Our proposal is similar to [18] in that it equally uses Criticality Aware Access Control. However our proposal includes a number of design choices specific to the IMD context and considers placement on a proxy device for personal security that were, so far, not considered in the design and implementation of CAAC framework. Also it proposes to use virtual spaces to dynamically allow access of a patient’s IMD to a medical practitioner only during emergencies. Components of our framework are as follows:

6.1 Role Management

The main functions are to assign proper roles to subjects and providing them appropriate privileges based on their roles according to subject-to-role, role-to-role, and role-to-privilege constraints. IMD Access roles are defined and assigned to a subject when it becomes a part of an IMD system of the patient and is based on his actual position and permissible actions.

Another type of role management happens by use of a Web Service using which a medical practitioner can join a virtual space based on his current location and can be contacted for providing emergency treatment to a patient with IMD is case of medical emergency when the medical practitioner with access privileges are not available as the patient is in some other area. A subject can have multiple space-roles, i.e. different roles in different spaces. The privileges are, however, only given to access the IMD within the space the subject inhabits when there is an emergency. Therefore system roles are a subset of space roles. The proxy on sensing a medical emergency, instead of failing open to make it accessible to all, will access the virtual space by specifying the kind of service needed for the IMD and its location to get a list of doctors who are available in the vicinity. With a doctor's consent it will allow access to the IMD for a limited period defined below to only doctor and not to all. The doctor will be provided credential and a complete log will be maintained.

Context information is used to keep track of changes that occur within the system to make Access Control decisions [17]. Context information is helpful in making the Proxy aware of a particular state in the patient or environment to determine which type of Access Control needs to be enforced.

6.2 Emergency State Management

Emergency State differs from a Normal State in following ways: 1) Emergency State requires automatic changes in access policies, unlike Normal State where contexts are evaluated on request, 2) all policy change with respect to Emergency State are temporary; 3) In Emergency State context non-readability must be ensured. Therefore when a patient is in Emergency State we require real time guarantees for mitigation.

6.3 Criticality

Criticality is a measure of the level of responsiveness in taking corrective actions to control the effects of a critical event and is used to determine the severity of critical events. To quantify this attribute, the term Window-of-Opportunity (Wo) is used in [18], which is an application dependent parameter defining the maximum delay that can possibly be allowed to take corrective action after the occurrence of a critical event. A Window-of-Opportunity = 0 indicates maximum criticality for a critical event while a Window-of-Opportunity = ∞ indicates no criticality.

In case of a critical event, the Proxy System needs to implement a new set of access policies which facilitate timely action called Emergency-Aware Access Policies (similar to CAAP) and during their execution, the system is said to be in the Emergency - mode. When the critical event is controlled, the system is removed from the Emergency - mode. If access policies are not changed during the onset of a critical event, then it may not be possible to control the critical event however diluting Access Control polities during critical events may introduce security concerns therefore duration of relaxation of Access Control policies needs to be managed carefully.

Criteria used for managing the Emergency mode: 1) the Window of opportunity Wo, 2) the time instant when the criticality is controlled T_{EOC} and 3) the time instant when all necessary actions to handle criticality has been taken (T_{EU}). The maximum duration for which the system can be in Emergency mode $T_{Emergency}$ is given by: $T_{Emergency}$ =min (Wo, T_{EU}, T_{EOC}). The Proxy continually determines the criticality level and on observing a critical event changes the access policies to Emergency-Aware and enters the Emergency State. If there is no criticality, then the system checks if it is in the EAAP - mode, if so, it returns the system to its Normal State and enforces the appropriate policies when access is requested.

Figure 2 below is a state transition diagram showing the Proxy States and figure 3 shows the Proxy architecture.

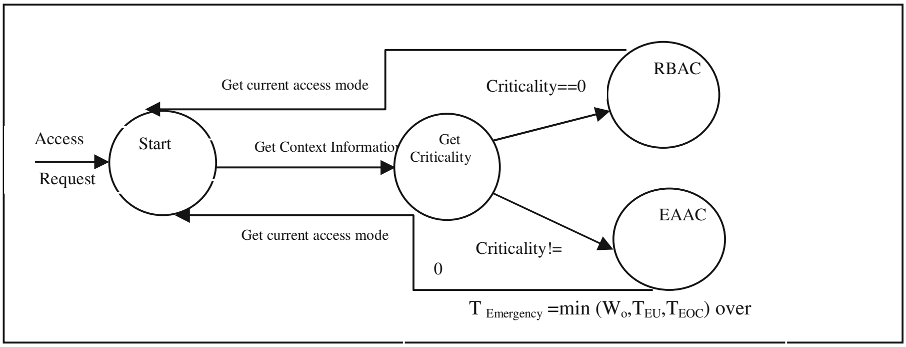


Fig. 2. State Transition Diagram for Emergency Aware Access Control using Proxy Device

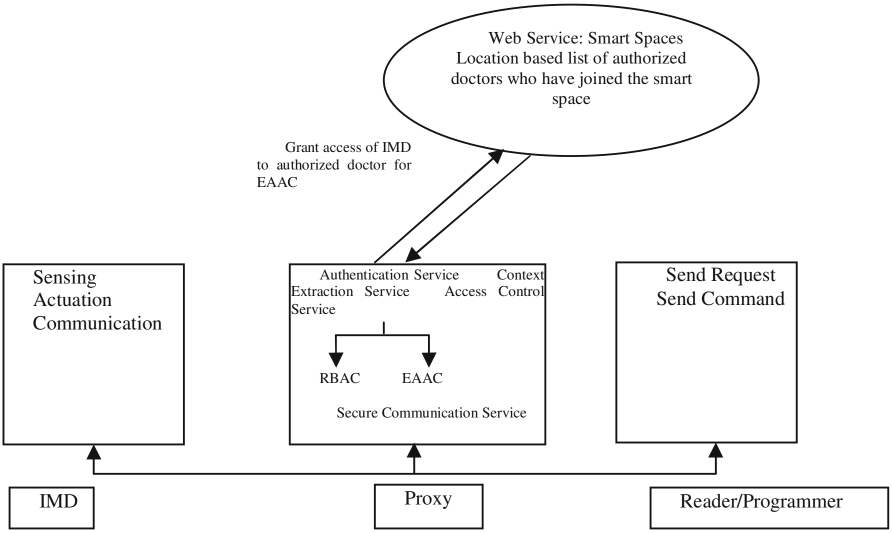

Web Service: Smart Spaces
Location based list of authorized doctors who have joined the smart space
Grant access of IMD to authorized doctor for EAAC
Sensing
Actuation
Communication
Authentication Service Context Extraction Service Access Control Service
RBAC
EAAC
Secure Communication Service
Send Request
Send Command
IMD
Proxy
Reader/Programmer

Fig. 3. The Proposed Proxy Architecture

The proxy implements RBAC [14] for Normal State and EAAC in Emergency State. The Criticality Management Unit of CAAC Framework [18] can be directly used here.

An Example

For a Cardiovascular implantable electronic device (CIED) the roles can be CIED physician, Clinically employed allied professional (CEAP) - group of nurses, physician assistants, technologists, technicians, and engineers, Heart failure (HF) care. Privileges can be: Device interrogation, Device programming. The proxy maintains a list of subjects that has access to the IMD when a critical event occurs and authorized staff is unavailable, it queries a web service to get doctors in the vicinity, after getting permission from a doctor it updates the Access Control for the IMD to make it accessible to the doctor and provides him temporary credentials. When criticality is over, policies are rolled back and RBAC is restored.

7 Conclusion

By emphasizing on security vulnerabilities of the IMD security system when fail open access is given in an emergency, we try to draw attention of researchers on the dangers of this aspect. We propose an Emergency Aware Access Control framework to deal with this important issue. The idea of controlled and not complete access is emphasized here where instead of giving fail-open access in emergency condition which may lead to security breach, an Emergency Aware Access Control technique is proposed which gives security even in emergency. Our Access Control runs on a proxy device and works in two modes: RBAC mode and EAAC mode based on context information extracted from the IMD and provides required security against unauthorized access in normal as well as emergency conditions.

References

1. Insulin pumps - global pipeline analysis, opportunity assessment and market forecasts to 2016. GlobalData, http://www.globaldata.com
2. Halperin, D., Heydt-Benjamin, T.S., Fu, K., Kohno, T., Maisel, W.H.: Security and privacy for Implantable medical devices. IEEE Pervasive Computing 7(1), 30–39 (2008)
3. Roberts, P.: Blind attack on wireless insulin pumps could deliver lethal dose. Threatpost (blog post) (October 2011), http://threatpost.com/en_us/blogs/blind-attack-wireless-insulin-pumps-could-deliver-lethal-dose-102711
4. Li, C., Raghunathan, A., Jha, N.K.: Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. In: Proceedings of the 13th IEEE International Conference on e-Health Networking, Applications, and Services, Healthcom 2011 (June 2011)
5. Burleson, W., Clark, S.S., Ransford, B., Fu, K.: Design challenges for secure implantable medical devices. In: Proceedings of the 49th Annual Design Automation Conference (DAC 2012), pp. 12–17. ACM, New York (2012)
6. Bergamasco, S., Bon, M., Inchingolo, P.: Medical data protection with a new generation of hardware authentication tokens. In: Mediterranean Conference on Medical and Biological Engineering and Computing (MEDICON), Pula, Croatia, pp. 82–85 (2001)
7. Schechter, S.: Security that is Meant to be Skin Deep: Using Ultraviolet Micropigmentation to Store Emergency-Access Keys for Implantable Medical Devices. In: USENIX Workshop on Health Security and Privacy (2010)
8. Rasmussen, K.B., Castelluccia, C., Heydt-Benjamin, T.S., Capkun, S.: Proximity-Based Access Control for Implantable Medical Devices. In: ACM Conference on Computer and Communications Sexscurity (2009)
9. Halperin, D., Heydt-Benjamin, T.S., Ransford, B., Clark, S.S., Defend, B., Morgan, W., Fu, K., Kohno, T., Maisel, W.H.: Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses. In: IEEE Symposium on Security and Privacy (2008)

10. Denning, T., Fu, K., Kohno, T.: Absence Makes the Heart Grow Fonder: New Directions for Implantable Medical Device Security. In: HotSec (2008)
11. Gollakota, S., Hassanieh, H., Ransford, B., Katabi, D., Fu, K.: They Can Hear Your Heartbeats: Noninvasive Security for Implanted Medical Devices. In: ACM SIGCOMM (2011)
12. Sandhu, R., Samarati, P.: Access control: Principles and practice. IEEE Communications Magazine 32(9), 40–48 (1994), http://www.list.gmu.edu/journals/commun/i94ac%28org%29.pdf
13. D. of Defense, Department of defense trusted computer system evaluation criteria, Department of Defense Standard, Tech. Rep., (December 1985), http://csrc.nist.gov/publications/history/dod85.pdf
14. Sandhu, R., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role Based Access Control Models. IEEE Computer, 38–47 (February 1996)
15. Covington, M.J., Long, W., Srinivasan, S.: Secure Context-Aware Applications Using Environmental Roles. In: Proc. of 6th ACM Symp. on Access Control Models Tech. (2001)
16. Al-Muhtadi, J., Ranganathan, A., Campbell, R.H., Mickunas, M.D.: Cerberus: A Context-Aware Security Scheme for Smart Spaces. In: Proc. IEEE Percom (2003)
17. Hu, J., Weaver, A.C.: Dynamic, Context-aware Security Infrastructure for Distributed Healthcare Applications. In: Proc. 1st Workshop on Pervasive Security, Privacy Trust (2004)
18. Gupta, S.K.S., Mukherjee, T., Venkatasubramanian, K.: Criticality Aware Access Control Model for Pervasive Applications. In: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM 2006), pp. 251–257. IEEE Computer Society, Washington, DC (2006)
19. USPTO Patent Application 20080044014. Secure telemetric link, http://www.freshpatents.com/Secure-telemetric-link-dt20080221ptan200800%44014.php?type=description
20. Venkatasubramanian, K., Gupta, S.: Security for pervasive healthcare. In: Security in Distributed, Grid, Mobile, and Pervasive Computing, pp. 349–366 (2007)
21. Cherukuri, S., Venkatasubramanian, K.K., Gupta, S.K.S.: Biosec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body. In: International Conference on Parallel Processing Workshops, pp. 432–439 (October 2003)
22. Harland, C.J., Clark, T.D., Prance, R.J.: Electric potential probes - new directions in the remote sensing of the human body. In: Measurement Science and Technology, vol. 13, p. 163 (2002)
23. Hansen, J.A., Hansen, N.M.: A taxonomy of vulnerabilities in implantable medical devices. In: Proceedings of the Second Annual Workshop on Security and Privacy in Medical and Home-care Systems (SPIMACS 2010), pp. 13–20. ACM, New York (2010)
24. Savci, H., Sula, A., Wang, Z., Dogan, N.S., Arvas, E.: MICS transceivers: regulatory standards and applications [medical implant communications service. In: Proceedings of IEEE SoutheastCon 2005, pp. 179–182 (April 2005)
25. Calero, J.M.A., Perez, G.M., Skarmeta, A.F.G.: Towards an Authorization Model for Distributed Systems based on the Semantic Web. IET Information Security. IET 4(4), 411–421 (2010)
26. Ni, Q., Bertino, E., Lobo, J., Calo, S.B.: Privacy aware Role Based Access Control. IEEE Security and Privacy 7(4), 35–43 (2009)

Index Page Based EDoS Attacks in Infrastructure Cloud

Bhavna Saini and Gaurav Somani

Central University of Rajasthan,
India
{bhavnasaini,gaurav}@curaj.ac.in

Abstract. One of the prominent attribute of cloud is pay-per-use, which can draw in the attackers to detriment the cloud users economically by an attack known as EDoS (Economic Denial of Sustainability) attack. This work identifies a novel class of attack in the area of EDoS attacks. Our focus is on defending the first page of any website i.e. Index Page. One of the important fact about index page attack, is that the index page of any website in this universe is available freely and even without any authentication credentials. To mitigate this attack and substantiate the difference between the legitimate and non-legitimate user, we have analyzed human behaviour of browsing and DARPA DDoS dataset. This analysis has helped us to design various models, ranging from strict to weak index page prevention models. The proposed schemes are implemented as a utility IPA-Defender (Index Page Attack Defender), which works well with minimal overhead and do not affect the legitimate users at all.

Keywords: Cloud Computing, Cloud Security, DDoS, EDoS, Index page.

1 Introduction

Cloud computing refers to a new business model which provides computational resources "as a service" [1]. These resources are modelled as infrastructure (IaaS) , software (SaaS) and platform (PaaS) and form a layered structure. The business like features of cloud such as pooling and elasticity of resources, on-demand service, multi-tenancy, costing and quality of service, all together are pulling in the users from both public and private sectors towards the cloud services. Due to cloud popularity many attacks are planned to degrade its performance and make cloud service unavailable for long time results in distraction of users from using cloud services. DDoS (Distributed Denial of Service) attack is one of the major and severe attacks which cause the denial of services to the user by flooding the target server with large number of packets, such that the server gets degraded and becomes unavailable to handle the further incoming requests [2]. However, in cloud environment DDoS attack cannot take place with ease, as the availability of on-demand resources exempt the server from getting downgraded. On the other side, the cloud user is impacted economically while serving these unsought

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 382–395, 2014.

requests from attacker. This variant attack is known as EDoS (Economic Denial of Sustainability)[3][4]. The basic intention of EDoS attacker is to make the cloud services undergo by sending fake requests and increasing the load in order to increase customers bill. The easygoingness of HTTP-GET requests pull in the attacker to go for it, even in case of slow attacks with small number of requests[5]. These slow and low attacks utilize good number of resources as well, which led to the additional unnecessary cost on user. J. Idziorek and M. Tannian [6] took Amazon EC2 pricing metrics as a reference and evaluated the cost of resources utilized, while serving the average size web pages on internet. The authors found that the average web page size is 320 KB and serving one HTTP-GET request per minute for a month will cost around $ 2.04 to the user. In the next section, the paper describes the problem statement.

1.1 Problem Statement

Web services are governed by HTTP as the base protocol and the business needs which are technology dependent are moving towards web service based applications. HTTP-GET flooding is a type of application layer DDoS attack which floods the server with large number of HTTP requests [2]. Sending HTTP-GET requests on the server is one of the most popular ways to retrieve and utilize server side resources which further result in extra cost addition to users bill. However, applying a bulk of HTTP-GET requests on a single page of a site, can also impact the servers performance.

This work tries to introduce to a new subclass of these attacks. The “index page” is the initial page of any web site and does not require any authentication or challenge response. Even, there is not a single website in the universe which requires any authentication to supply the index page. Employing bulky and concurrent HTTP-GET requests generate large amount of resource consumption overhead on server. We call these attacks as “index page” attacks and can be applied to any website in this universe.

Index-page attacks may become one of the most critical problems for cloud users. Even the slow attacks with small number of requests, can utilize adequate resources, which results in increase of response time for the legitimate users as well as impact economically while serving these unsought HTTP-GET requests from attacker. In this case of slow and small attack where number of requests is not much, the server does not get downgraded and not even stops working. Increase in response time and drop off in number of actual users may not be sufficient for some attackers who have intention to strike down the server totally.

To harm the server to a very large extent, the attacker will typically increase the number of bots and the number of requests sent at a time. The attacker will try to utilize more and more resources such that the server utilization reaches up to 100% and stops working. In cloud, even these large numbers of requests will not be capable of downgrading the server because of the easy availability of resources as and when required. This saves the server from getting halted and keep it in up and running condition, but adds on the extra monetary value to the users invoice for the additional resources. If this type of attack takes place continuously, it will add resources as well as price to a very large extent to the cloud users account. This excessive usage can make it very difficult for the consumer to sustain economically in the market and led to the condition known as Economic Denial of sustainability (EDoS). EDoS aims to knock down the server in terms of monetary values by consuming resources as much as possible. It is really difficult to find out the limit of the requests at which the EDoS attack will happen. EDoS attack depends upon the server configuration, resources available with the cloud users and the affordability of the resources. To have intense learning of EDoS, an example of "Amazon EC2 standard on-demand instance" is taken. The cost to run this instance on Linux/Unix is $0.060 per hour [8]. On arrival of large number of requests, the server will ask for addition of one more instance to fulfill these excess requests. This result in addition of an extra cost to user's bill i.e. $0.060. The continuous increase in requests will result in continuous demand of resources and at one point EDoS will occur.

The DDoS attack can also happen on "index page" in cloud, as if the number and efficiency of HTTP-GET requests increases. The attackers employ a large number of requests for a very long time which increase the need for more and more resources continuously. This continuous need of extra resources can make it unaffordable for users which result in cause of DDoS.

This paper is also anticipating that even the ethically wrong service provider may itself plan such type of attacks on the websites present in cloud, as they raises profit by providing the resources to users, consequently more the resources utilized, more the profit provider gets out of it.

Usually website employs multiple methods to investigate issues such as authentication, challenge response protocol, so that any few of the users can be allowed to use the resources and not all of them. Additionally, there are ways by which web servers are blocking the requests which are coming to the web server, on the basis of URLs and IP ranges [10].

Our work concentrates on a different problem where these solutions do not perform at all, as the differentiation between attacker and actual user is really very challenging and difficult. In order to not get detected, the attacker sends the requests with same structure and semantic as legitimate users [9], which makes it really difficult to distinguish between them. "Index page" is our area of concern as it is free of cost and does not require any authentication. All the attacks mentioned above i.e. Slow HTTP-GET attacks, EDoS and DDoS can take place on the index page with great ease, as for most of the websites in the world this page is freely available.

Usually these index pages are sized between few hundred KBs to MBs. Various popular public sites have images, videos, advertisements and diverse multimedia items on the front page. Some of the best examples of this type of sites include any online audio, video sites, online shopping sites, news channel, social networking sites and many more for ex. Yahoo, facebook, Gmail, ebay etc. The additional request done by the browser to load such dynamic multimedia items, adds a lot to the effort of loading the page. Hence with the increase in page size, the page load time on client side has been also increased.

2 Web Pages Categorisation

The index pages are categorized into static and dynamic. The static index page remains the same until and unless it is changed manually and takes less time than dynamic page to deliver the content. On the other hand, the content of dynamic pages gets generated at the time of request; it displays the viewer specific content or access the database for different responses. As compared to static pages, the dynamic pages are more resource-intensive and require more effort and time to fulfil a request. The static and dynamic index pages are further categorised on the basis of common attributes they share.

1 **Free Availability of Index Page with No Authentication for Entrance**

These Index pages costs nothing for the user and are available without any credentials. Online news, portals, multimedia sites are some of the best examples which define this category very well. All these pages centre on the content of the page and are available for every single user without any credentials. Along with index page, all the other pages of these sites are also accessible to every individual and are free of cost.

2 **Free Availability of Index Page with Authentication Based Entrance**

Under this category, the one and only page, which is free for user is the very first page of the site i.e. index page. The page provides the user with the login option, which is required to access rest all the pages of the site.For ex. Facebook, Gmail, Yahoo mail and many more examples are present, where the user can access only the index page until and unless login is done. On providing the credentials, the user can easily move to other pages of the site.

3 **Free Availability of Index Page with Optional Authentication Based Entrance**

This category can be defined as a combination of the categories defined above. Here some of the pages are freely available, while some require authentication for example Youtube. This is a multimedia site where various videos are present on index page which are freely accessible, while some need login credentials.

3 Related Work

The work done for DDoS defense includes various mechanisms which takes place at TCP, IP or Application layer according to the attributes or features used for analysing and detecting the attacks.

Entropy based approaches had been used to detect the DDoS attacks [10], where TCP flow rate calculated to identify the normal and malicious packets. An advanced scheme which performs better than prior entropy approaches was introduced, which differentiates the legal traffic and flash crowd from Low rate

DDoS [11]. The weak point of this advanced entropy based approach is the longer response time.

S. Renuka devi and P. Yogesh [12] proposed a DDoS detection method based on the web browsing behaviour of a user and defending it using a rate limiter and a scheduler. A distributed filtering technique using overlay networks was proposed by Zhang fu [1], where lightweight authenticators are used for filtering. This method proves helpful in protecting the individual servers from bandwidth flooding but is expensive and impractical for specific application channel. The author mitigates the problem by proposing Port-hopping method. [13]. Another approach was proposed depending on the HTTP-GET request arrival rate [14], which give satisfactory results.

Yi Xie and Shun-Zheng Yu [15] proposed a method based on document popularity in which anomaly detector based on hidden semi-Markov model is used to detect the attacks. The biggest problem of Hidden Semi-Markov method was the algorithm complexity. Wei-zhou lu and shun-zheng yu used the Hidden semi markov model (HSMM) [16] to describe the web browsing patterns and detecting flooding attack. The web browsing pattern used to differentiate between legitimate and unauthorized user.

D-WARD, a self directed DDoS detection and defense system was proposed that works at source end [2]. For application layer DDoS such as HTTP flood, DWARD does not prove efficient as it imposes large overhead on the routers and is less accurate in differentiating legitimate user and attacker.

A high performance DDoS prevention method named as GESNIC (Gigabit Ethernet Secure Network Interface Controller) was proposed in [17]. In this paper, a table is maintained containing IP and URI content hash of incoming GET packets and comparison is done against new incoming packets. If collision occurs the packet is dropped.

One of the promising approaches against DDoS and EDoS was using challenge responses for distinguishing between legitimate and attack requests. M. Sqalli et.al. verified the incoming requests on the basis of graphic turing tests [3]. This technique helped in mitigating the EDoS attacks on cloud. Further an enhanced EDoS-shield approach was proposed to stop EDoS attack coming from spoofed IP addresses by taking TTL(Time to live) value present in IP header [4]. Authentication using CAPTCHA proves very effective against HTTP flood attack but the Turing test led to annoy the user which are coming from spoofed IP addresses by taking TTL(Time to live) value present in IP header.

Few techniques were proposed to detect the fraudulent resources consumption in public cloud environment. The detection methodology make use of zipfs law and entropy computation where the slopes for training and test data set are generated using zipfs law and compared for the differences of each user session lengths and comparing the result to the standard [6].

The technique proposed is different from previous work as the focus is only on defense of index page. So far there is no work which concentrates or provides solutions for the novel attack we have discovered. All the techniques which we have discovered so far in this section provides solutions for DDoS or EDoS attacks with the help of challenge response techniques [18] or their techniques are so generic or are not specific to cloud environment. Due to the features like on-demand and elastic resources DDoS attack is not applicable to the cloud in its pure form. To protect the Home page from EDoS in cloud, various models have been proposed. These models can also work for other pages of websites.

4 Initial Set of Experiments

Some experiments were performed initially to find out the severity level of HTTP-GET attacks on index pages of various websites.

4.1 Analysis of Severity Level of Attack on Index Page

To check the severity of HTTP-GET attack on index page an experiment was carried out to find the CPU utilization when a large number of bulky and concurrent requests are transmitted to "index page" as shown in figure 1. An Apache web server was installed and configured along with an index page. A large number of simultaneous requests are sent to the server using *"apache benchmark ab"* utility through attacker machine to get the index page. For these incoming requests CPU utilization is also calculated.

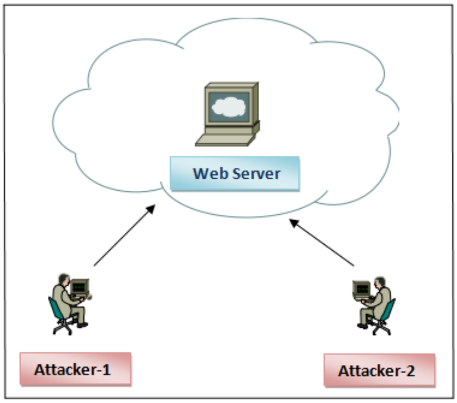


Fig. 1. Experimental Setup

Experimental Setup. An Intel core i7-3632QM, 2.20 GHz CPU, 4GB RAM system has been used for server setup. As index page attacks make more sense in cloud environment, hence this experiment is performed on virtual machine (VM). VM running on hypervisor, which resembles with an infrastructure as a service(IaaS) cloud scenario. Even the resources given to the VM which is running as web service, have been varied to show the impact of attack on "on-demand" resource allocation in cloud as shown in Table 1. As far as the IaaS are concerned, the resource allocation is governed by a Service Level Agreement (SLA) and these SLAs usually define the Min. and Max. resources available to a specific VM instance deployed inside an infrastructure cloud. A typical SLA may detail about specific policy of resource allocation. These policies of automatic and on-demand scaling and shrinking are termed as auto scaling in few of the infrastructure products. For ex. If a virtual machine CPU utilization is in overloaded condition (85%) for specific duration of time say 3 minutes, in that case this machine requires more CPU. On the other hand, if it has underload condition (30%) for the same duration i.e. 3 minutes, then some extra CPU are removed from the machine.

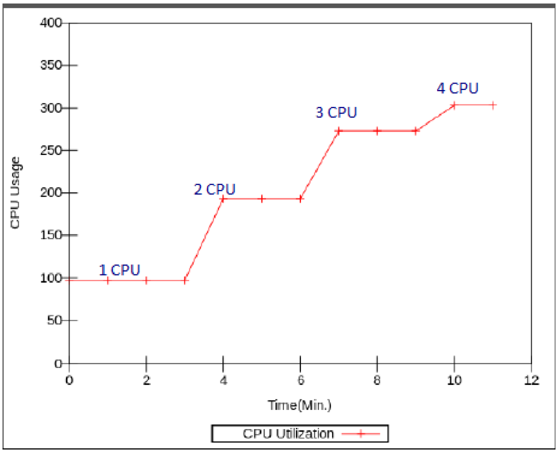


Fig. 2. CPU utilization with different number of CPUs

Results and Discussion. To interpret the auto-scaling feature, an experiment is performed. Initially, only one CPU is allocated to the VM which is running as web server. Around 10000 requests with 200 concurrency are sent from two attacker machines to get the index page. While serving these large number of bulky and concurrent requests, 90-99% of the CPU is utilized as shown in Figure 2 and reported in Table 1. This high utilization occurs continuously for 3 minutes, which is a signal for CPU increment. Therefore, one more CPU is hot-plugged in the VM and the utilization is observed for next 3 minutes. These incoming requests are large enough that even on increasing CPU, it is still overwhelming 90-99%. This results in addition to one more CPU to the machine. The add-on of CPU brings down the utilization to around 80-90%, but as per the auto scaling policy defined above, 85% is the overloaded condition. Hence, going with the rules, one more CPU is allocated to the server. The utilization has now come to a stable state i.e. in between 60-70%. The results of the experiment show that in cloud environment, DDoS attacks do not really occur. The on-demand resource utility prevents DDoS, but adds an extra cost to customer's bill. This economic loss due to unwanted requests can lead to EDoS i.e Economic denial of sustainability.

Table 1. CPU Utilization

Host-CPU	Guest-CPU	No. of requests	Concurrency	Avg. Utilization(%)
1	1	10000	200	98
2	2	10000	200	97
3	3	10000	200	82
4	4	10000	200	73

5 Human Behaviour and Index Page Attacks

Our work aimed at understanding the universal web browsing and navigation behaviour. In past, analysis of browsing behaviour has been done for different web pages of the sites [19]. The work done in past didn't work for specific pages. Index page is our main area of concern, hence the human behaviour analysis is done for it.

To understand and evaluate the human behaviour, a survey is done with 50 internet users. This survey aimed at finding the number of revisits or refreshes of index page, a legitimate user can do in a specific interval of time. For this, a set of index pages of some of the most popular sites are used, as shown in table 2. Each and every individual is asked for the total number of refreshes done in a minute for these index pages. This survey was conducted by letting the participant know about the aim of our survey i.e. to know the maximum number of times a user can request the index page of a specific site. Even one care has been taken into account, that a user sends request only when first gets completed. All the requests are in serial manner.

Experiment which we are conducting gives us a criterion to differentiate between legitimate and non-legitimate users. Even the modern browsers/web-clients are capable enough to start multiple tabs or request the same page at the same time. This is also kept in mind while designing the EDoS prevention method in next chapter.

The outcome of the survey is interesting as shown in Table 2. The number of times an index page can be reloaded varies from site to site. For lightweight pages such as Facebook, the count is quite good, while on the other hand, for the sites where a lot of content is present on the page, such as yahoo the count goes less.

If we go by a model on per minute basis, a possibility is that an attacker can come to know the threshold value which differentiates between legitimate and non-legitimate users. Knowing this threshold, an attacker may plan for a DDoS, by sending requests from various systems simultaneously. Even on sending the requests equal to that of threshold, the server can be harmed when these requests are sent all together which is outside our scope of work. To verify the results obtained from survey, we also looked up for datasets related to DDoS. NASA traces [20] and DARPA datasets [21] were utilized by some of the authors in past for DDoS analysis [7], while some has built their own dataset by collecting the traces on web server. No latest DDoS datasets are available which could help us in our work. Therefore, we have used the DARPA dataset. The dataset collected for 1 week and contains the normal traffic data which shows the webpages accessibility behaviour of user. DARPA gives us an idea of the number of times a site is accessed by a single user in a time interval. The minimum count for accessing a site in a day is 1 and the maximum count is 70. As compared to the weeks data, the requests per minute from a user will be very low. The maximum number of times a legitimate user can access a web page in a minute will not be more than 10 requests. This gives a rise to our differentiation of legitimate users and non-legitimate users on the basis of human browsing behaviour. Even if the datasets are available where the EDoS attacks presence is there, it is not going to help much as the analysis and our focus is mostly dependent upon human browsing behaviour.

Table 2. Index Page Statistics

Index Pages	**Max. Value**	**Min. Value**	**Avg. Value**	**Std. Dev**
Facebook	34	16	23	5.9
Yahoo	7	5	6	0.6
Google	18	8	12	3.01
Ebay	16	7	11	2.54
Youtube	17	8	13	2.9
Wordtopdf	2	1	1	0.51

6 Proposed Models

To protect the index page from such unwanted requests, various models are proposed. Our models work for concurrent requests also. All these models form a spectrum which moves from strict to weak or vice-versa. The models provide a good flexibility to user in terms of working, according to the requirements.

6.1 Strict Model

As the name indicates, this model follows the strict rules to serve the index page. According to this model the page will be served only once. Some of the practical examples where this model can be applied include various mobile or desktop applications such as Gtalk, Facebook, WhatsApp and many more. All these applications have their index pages locally stored in the devices. Therefore, they don't need to call it every time. This locally stored page feature allows the user to access the page, even in case of "no network connection". In strict model, there is no possibility of DDoS or EDoS attacks, but availability issue can occurs as it serves only one time. This availability issue make it really hard for applications to adopt it in practical use. To get by this availability issue, a moderate model is proposed.

6.2 Moderate Model

The human web browsing behaviour analysis done in section 5 acts as a groundwork for this model. This analysis helps in directing the implementation of our Moderate model. The model provides the flexibility to alter the threshold for page count and time interval. This variability results in versatile models which are applied according to the demand. To get it on with this model, various sub models are defined. These sub models work with different values of page count and time interval. Lots of real examples are present, where these models can work effectively and efficiently. Gmail, Facebook are some of the best examples, where a legitimate user will access the home page only for few times within a time interval. Following the results of the survey in table 2, the model defines "n" as the maximum number of requests in a minute. Here "n" may be 34 for Facebook and 18 for Google. Requests exceeding the defined threshold will be dropped. Another category of sites are the one providing online services such as online file format converter. Some of these sites have conversion option on index page only. If an actual user is trying to use such services, it will take some time to finish off the processing and providing results. In that case, the chances are that the user will access the page at most two time in a minute. On the contrary, an attacker will send the requests only to bring forth the page, not to use up the services. In this scenario, the time based moderate model can be applied. According to our proposed model, only one request will be served within a minute. This model can stop irrelevant requests coming to the home page. Likewise the above defined model, one more moderate model is proposed which provides adaptability in both the page count as well as time interval. Here the various configurations defined are 1 request/3 minutes, 10 requests/3 minutes and 16 requests/minute. All these models will not impact the genuine user, but will limit the attackers. The user can choose these sub models as per the demand. As per our survey results, the sub model with 16 requests/minute can be employed on those sites, which are accessed on day-to-day basis such as online newspaper sites, shopping sites etc.

Unlike other approaches, our approach does not give any false positives, as a suspected attack may be removed by adding an additional count of "K" in the threshold appearing in the human browsing behaviour. This "K" may encompass, some of the other issues like auto querying by web services or synchronizing feeds continuously. This "K" can be decided by doing a dry run of those kind of services and can be added to the threshold.

6.3 Weak Model

The Weak model permits each and every incoming request. This is same as, Apache working without any measures against attacks.

7 Implementation and Results

Initially Mod-evasive, which is an evasive module of apache was used for implementation of proposed models. Mod-evasive works to protect against attacks such as HTTP-DoS, DDoS and brute force. While implementing the proposed models using mod-evasive, the results obtained were not satisfactory. The two main problems which occurred are :

1 *High CPU utilization while forbidding requests*
Mod-evasive results in very high CPU utilization, while forbidding the large number of concurrent requests. It works with 90-100% utilization, which is almost equal to the utilization that occurs while serving these large number of requests.

2 *Variation in output result*
Using Mod-evasive module, variation occur in the defined threshold value for page count and the actual results. When strict model was implemented using Mod-evasive, the threshold for page count was defined as 1 time. On the contrary, when experiment was performed, variations occurred in actual results. Instead of serving the page only once, it is serving more than one time. The same observations occurred when the moderate model was implemented using mod-evasive.

Further the models were implemented using a tool named as IPA-Defender, which also solves the problems occurred with Mod-Evasive.

7.1 IPA-Defender

IPA-Defender. (Index Page Attack Defender) is the tool implementing our proposed models. It protects various websites from index page based EDoS attacks. The implementation is carried out using Iptables. Iptables stand for "administration tool for IPv4 packet filtering and NAT" [22] . It is a command line tool that is provided by Linux kernel. According to the IPA-Defender algorithm, each and every request which is coming for index page is checked against the Iptables rules. Page is served to the user, until the threshold for its page count reaches out. Once it attains the doorway, it starts dropping all the connections coming from same IP address. This IP remains blocked for a specific period of time. On completion of this time interval, the IP is removed from temporary blacklist and allow the request from it.

Algorithm 1. IPA-Defender Algorithm

```
Require: Request R
Ensure: SERVE or DROP
procedure IPA-DEFENDER(R)
    while R do                                            ▷ New incoming request
        if $R_{Dest}$ = "Server-IP\index.html" then
            if $R_{IP}$ is in DROPLIST then
                DROP
                UNBLOCK the IP after TI            ▷ TI = Threshold for Time-Interval
                Exit
            else if $R_{IP}$ is in TEMPLIST then
                $N_{IP} \leftarrow N_{IP} + 1$            ▷ $N_{IP}$ = Number of pages requested
                if $N_{IP} <=$ MaxReqAllowed then   ▷ MaxReqAllowed = Threshold for Index Page to be Served
                    $R_{IP}$ is SERVE
                    Exit
                else $R_{IP}$ is DROP
                    DROPLIST $\leftarrow R_{IP}$                               ▷ Unavailable
                    Exit
                end if
            else $R_{IP}$ is SERVE
                TEMPLIST $\leftarrow R_{IP}$                                   ▷ Available
            end if
        end if
    end while
end procedure
```

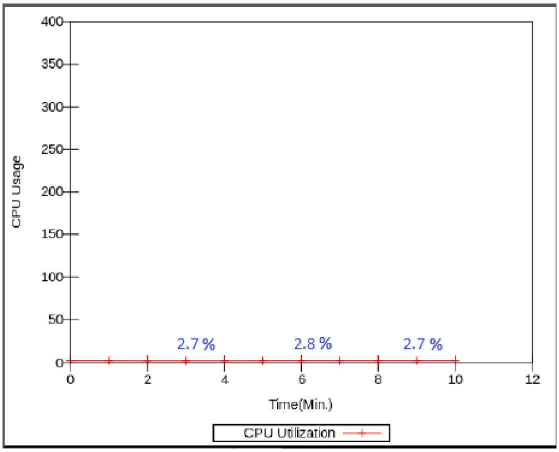


Fig. 3. CPU Utilization of IPA-Defender

From the point of view of CPU utilization, it is working very efficiently. As mentioned above in figure 2, for serving a large number of unwanted requests, the user has to increase the number of CPUs. The addition of such extra resources turns to be pointless for the user. The IPA-Defender has removed the need of additional resources. It has reduced the CPU utilization to 3% from 97%, which takes away the possibility of EDoS or DDoS totally as shown in figure 3.

8 Conclusion

The day by day increasing trend of cloud computing is due to its features like on-demand resources, pay-per-use, multi tenancy and many more. Besides these features, various issues are also associated. EDoS has become one of the most popular attacks in cloud, which occurs due to unwanted requests coming from attackers and it also adds a lot to the customers bill. In past, various techniques have been proposed to prevent this attack, but our work is quiet different. In this work, we focus on identification, prevention and detection of a novel EDoS attack i.e. **"Index page based attack"**. Index page is the cost and credential free page of any website, which can be easily accessed by any person in the world. Taking advantage of this, an attacker can send a large number of requests to index page, which further results in extra demand of resources due to on-demand nature of cloud (It could have been a DDoS attack if flexible resources are not there). These extra resources harm the user economically by adding up extra cost for the unwanted resources. Initially, experiments were performed for analysing severity level of attack on index page along with a survey for human web browsing behaviour analysis. Based on this, various models has been proposed which work with different threshold values of page count and time interval. The threshold value limit the number of requests and stop the unwanted requests. This technique is termed as "IPA-Defender", which provides very less overhead and positively stops the non-legitimate users without EDoS.

References

1. Fu, Z., Papatriantafilou, M.: Off the wall: Lightweight distributed filtering to mitigate distributed denial of service attacks. In: 2012 IEEE 31st Symposium on Reliable Distributed Systems (SRDS), pp. 207–212 (2012)
2. Beitollahi, H., Deconinck, G.: Analyzing well-known countermeasures against distributed denial of service attacks. Computer Communications 35(11), 1312–1332 (2012), `http://www.sciencedirect.com/science/article/pii/S0140366412001211`
3. Sqalli, M., Al-Haidari, F., Salah, K.: Edos-shield - a two-steps mitigation technique against edos attacks in cloud computing. In: 2011 Fourth IEEE International Conference on Utility and Cloud Computing (UCC), pp. 49–56 (2011)
4. Al-Haidari, F., Sqalli, M., Salah, K.: Enhanced edos-shield for mitigating edos attacks originating from spoofed ip addresses. In: 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1167–1174 (2012)

5. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM 53(4), 50–58 (2010), http://doi.acm.org/10.1145/1721654.1721672
6. Idziorek, J., Tannian, M.: Exploiting cloud utility models for profit and ruin. In: 2011 IEEE International Conference on Cloud Computing (CLOUD), pp. 33–40 (2011)
7. Okuhara, M., Shiozaki, T., Suzuki, T.: Security architecture for cloud computing. Fujitsu Sci. Tech. J. 46(4), 397–402 (2010)
8. Amazon-EC2, http://aws.amazon.com/ec2/
9. Ye, C., Zheng, K.: Detection of application layer distributed denial of service. In: 2011 International Conference on Computer Science and Network Technology (ICCSNT), vol. 1, pp. 310–314 (2011)
10. Kashyap, B., Jena, S.: Ddos attack detection and attacker identification. International Journal of Computer Applications 42(1) (2012)
11. Zhang, J., Qin, Z., Ou, L., Jiang, P., Liu, J., Liu, A.: An advanced entropy-based ddos detection scheme. In: International Conference on Information Networking and Automation (ICINA), vol. 2, pp. V2-67–V2-71 (2010)
12. Devi, S.R., Yogesh, P.: Detection of application layer ddos attacks using information theory based metrics (2012)
13. Fu, Z., Papatriantafilou, M., Tsigas, P.: Mitigating distributed denial of service attacks in multiparty applications in the presence of clock drifts. In: IEEE Symposium on Reliable Distributed Systems, SRDS 2008, pp. 63–72 (2008)
14. Das, D., Sharma, U., Bhattacharyya, D.K.: Detection of http flooding attacks in multiple scenarios. In: Proceedings of the 2011 International Conference on Communication, Computing &#38; Security, ICCCS 2011, pp. 517–522. ACM, New York (2011), http://doi.acm.org/10.1145/1947940.1948047
15. Xie, Y., Zheng Yu, S.: Monitoring the application-layer ddos attacks for popular websites. IEEE/ACM Transactions on Networking 17(1), 15–25 (2009)
16. Lu, W.-Z., Zheng Yu, S.: An http flooding detection method based on browser behavior. In: 2006 International Conference on Computational Intelligence and Security, vol. 2, pp. 1151–1154 (2006)
17. Kim, H., Kim, B., Kim, D., Kim, I.-K., Chung, T.-M.: Implementation of GESNIC for web server protection against HTTP GET flooding attacks. In: Lee, D.H., Yung, M. (eds.) WISA 2012. LNCS, vol. 7690, pp. 285–295. Springer, Heidelberg (2012)
18. von Ahn, L., Blum, M., Langford, J.: Telling humans and computers apart automatically. Commun. ACM 47(2), 56–60 (2004), http://doi.acm.org/10.1145/966389.966390
19. Kumar, R., Tomkins, A.: A characterization of online browsing behavior. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 561–570. ACM, New York (2010), http://doi.acm.org/10.1145/1772690.1772748
20. NASA-Server-Traces, http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html
21. DARPA-Dataset, http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/1999/training/week1/index.html
22. Iptable, http://linux.die.net/man/8/iptables

Privacy Preserving Ranked Keyword Search over Encrypted Cloud Data

Dinesh Nepolean, I. Karthik, Mu. Preethi, Rahul Goyal, and M. Kathirvel Vanethi

Amrita Vishwa Vidyapeetham, Coimbatore, Tamilnadu, India
{smnepolean,karganesh93,mupreethi,
goyal.1234rahul,vanethikathirvel}@gmail.com

Abstract. This paper presents a scheme that discusses secure rank based keyword search over an encrypted cloud data. The data that has to be outsourced is encrypted using symmetric encryption algorithm for data confidentiality. The index file of the keyword set that can be searched is outsourced to the local trusted server where the keyword set that is generated from the data files is also stored. Unlike architectures proposed in previous papers where encrypted index is stored in un-trusted server, in our schema the index is stored in the local trusted server to reduce round trip time and processing overhead. This is done so that the un-trusted server cannot learn about the data with the help of the index formed. The index is created with the help of Aho-Corasick multiple string matching algorithm which matches the pre-defined set of keywords with information in the data files to index them and store relevant data in B+ trees. Whenever the user searches for a keyword, the request is sent to the local trusted server and the indexed data is referred. The files are retrieved and ranked based on certain relevance criteria. The parameters required for ranking is got from the data stored while indexing.

Keywords: Symmetric Encryption algorithm, Rank based search, multiple string matching, relevance scoring, privacy preserving, and cloud computing.

1 Introduction

Cloud Computing is the evolving technology that has changed the way of computing in IT Enterprise. It brings the software and data to the centralized data centers from where a large community of users can access information on pay per use basis. This poses security threats over the data stored. Data confidentiality may be compromised which has to be taken care of. So it becomes necessary to encrypt the data before outsourcing it to the cloud server. This makes data utilization a challenging task. Traditional searching mechanisms provide Boolean search to search over encrypted data, which is not applicable when the number of users and the number of data files stored in the cloud is large. They also impose two major issues, one being the post-processing that has to be done by the users to find the relevant document in need and the other is the network traffic that is undesirable in present scenario when all the files matching with keywords is retrieved. But this paper proposes ranked keyword search that overcomes these issues.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 396–403, 2014.

The paper is formulated as follows. The related work is summarized in Section 2. The proposed system and architecture diagram is covered in Section 3. The scheme is split into encryption module, string matching module, indexing module and ranking module which are also discussed under Section 3. Section 4 gives the gist about future ideas and proposals.

2 Related Works

It is an important research problem to enable the cloud service provider to efficiently search the keyword in encrypted form on encrypted files and provide user data privacy at the same time. We have read the following papers.

2.1 Practical Technique for Search over Encrypted Cloud Data

This paper discusses on sequential scanning search technique [1] that searches over encrypted data stored in cloud without losing data confidentiality. The technique is provably secure and isolates the query result whereby the server doesn't know anything other the search result. It also supports functionalities such as controlled searching by server, hidden query support for user which searches for a word without revealing it to the server. With searchable symmetric encryption [7] and pseudorandom sequence generating mechanisms that are secure, encrypted data can be effectively scanned and searched without losing data privacy. The scheme that is proposed is flexible that it can be further extended to support search queries that are combined with Boolean operators, proximity queries, queries that contain regular expression, checking for keyword presence and so on. But, in case of large documents and scenarios that demand huge volumes of storage, the technique has high time complexity.

2.2 Public Key Encryption with Keyword Search

Dan Boneh proposed a solution for searching over the cloud data that is encrypted using the Public key Crypto System [2]. The idea is to securely attach or tag the related keywords along with the each file. This will avoid the need to completely decrypt the file and save the time of scanning entire file to check if the keyword exists. The file is encrypted using a public key encryption algorithm [2] and the keywords are encrypted by PEKS algorithm. To retrieve the document containing keyword W, send only the Trapdoor (W) to server. He proposed two methods for construction of this scheme, one using the bilinear maps and other using Jacobi symbols. The problem with this scheme is that every tag of all the files has to be processed for finding the match.

2.3 Boolean Symmetric Searchable Encryption

Most of the techniques discussed so far focused only on single keyword matching but in real-time scenarios users may enter more than one word. Tarik Moataz came up with a solution to tackle such challenges of searching multiple keywords over the encrypted cloud data. The construction of Boolean Symmetric Searchable Encryption (BSSE) [11] is mainly based on the orthogonalization of the keyword field according to the Gram-Schmidt process. The basic Boolean operations are: the disjunction, the conjunction and the negation.

2.4 Fuzzy Keyword Search

The traditional searching techniques retrieve files based on exact keyword match only but Fuzzy keyword search technique extends this feature by supporting common typos and format inconsistencies that occurs when the user types the keywords. The data privacy that is maintained during exact keyword search is ensured when this method is used. Wild card based technique [4] is used to create efficient fuzzy keyword sets that are used for matching relevant documents. The keyword sets are created using Edit Distance algorithm that quantifies word similarity. These keyword sets reduce storage and representation overhead by eliminating the need to generate all fuzzy keywords, rather generating on similarity basis. The search result that is provided is based on a fuzzy keyword data set that is generated whenever the exact match search fails.

2.5 Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data

Traditional searchable encryption techniques support Boolean search to search over the encrypted data which is not efficient when huge number of data files and large documents are stored in the server. The search result requires post processing to find the files of interest and since large numbers of files are retrieved on a presence or absence of keyword basis, there is unnecessary network traffic. The scheme proposed through this paper overcomes these by ranked search mechanism. Scoring mechanism that is used in information retrieval quantifies and rank-orders the files in response to a search query. The quantification is done based on the keyword frequency relevance criteria. This is used to set up scores for the files and rank them. When the user wants to search for a word, say w, he is authorized to create a trapdoor T_w and send it to the server. The index is searched and the corresponding files are ranked and retrieved which increases the file retrieval accuracy. When an optional value, say k, is sent, the top k-most relevant files are retrieved.

3 Proposed System

We have proposed an efficient scheme which enables the Cloud Service Provider (CSP) to determine the files that are related to the keywords searched by the user, rank them and send the most relevant files without knowing any information about the cloud. Our schema consists of three entities: Data owner, Un-trusted cloud server and local trusted server. The data owner is the one whose data is stored in cloud server and he is also authorized to search over his files. Cloud server is an un-trusted server which provides storage service where data owners store their documents in encrypted form. The trusted local server stores the index that is created for the files. The system architecture is shown in Fig 1. We assume that authorization of users and keys used for encryption are managed by the local trusted server.

Notations:

1. C (F1, F2, .., Fn) : Files to be uploaded in cloud server.
2. W (w1, w2, ..,wi) : Keywords extracted from C.

3.1 System Architecture

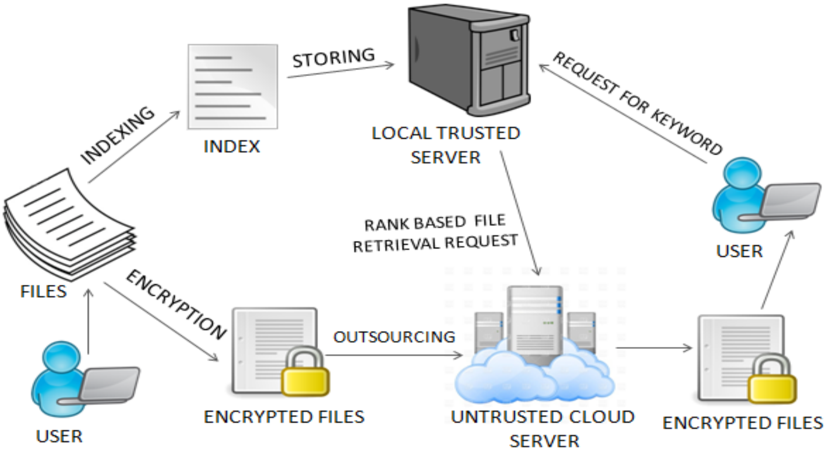


Fig. 1. System Architecture

3.2 Encryption Algorithm

In order to overcome security threats, the data files are encrypted using a symmetric encryption algorithm AES (Advanced Encryption Standard). The encryption process followed is as follows:

1. The data file that has to be encrypted undergoes four phases which is repeated n times (n being decided based on the level of security required).

   The phases to encrypt the file:

   1. Add round key: 128 bit sequence of the input sequence is XORed with the 128 bit expanded key obtained from key expansion (mentioned later) undergone by the key k.
   2. Sub Bytes Transformation: Each bit of the 128 bit sequence input is substituted by another bit that is pre-defined with the help of a look up table.

   3. Shift Row: The 128 bit input is arranged in a matrix format and every row of the matrix is shifted.
   4. Mix column Transformation: Each column of the input matrix that is formed is substituted by another column.
2. The key, k for encryption is expanded by Key Expansion technique to be used in a phase mentioned above. The key expansion routine, as part of the overall AES algorithm, takes an input key of 4*N_k bytes, or N_k 32-bit words. N_k has value 4, 6, or 8. The output is an expanded key of 4*N_b*(N_r+1) bytes, where N_b is always 4 and N_r is the number of rounds in the algorithm.

3.3 String Matching Algorithm

Aho-Corasick is found to be the efficient algorithm for multiple string matching that finds all occurrences of the pattern present in the files that are to be outsourced to the un-trusted cloud server.

The algorithm consists of two parts:

The first part is building of the tree (trie) from keywords we want to search for, and the second part is searching the test for the keywords using the previously built tree. The tree is a finite state machine, which is a deterministic model of behavior composed of finite number of states and transitions between those states. In the first phase of tree building, keywords are added to the tree where the root node is just a place holder and contains links to other letters. A trie is the keyword tree for a set of keywords K is a rooted tree T such that each edge of T is labeled by a character and any two edges out of a node have different labels.

CONSTRUCTION for set of keywords W = {W1, … Wk} and n = ∑= |Wi|.

Begin with the root node

Insert each keyword W, one after the other as follows:

Starting at the root, follow the path labeled by characters of $W_{i:}$

- If the path ends before Wi, continue it by adding new edges and nodes for the remaining characters of Wi
- Store identifier i of Wi at the terminal node of the path. This takes clearly O(|W1| + … + |Wk|) = O(n) time

LOOKUP of a string P:

Starting at root, follow the path labeled by characters of P as long as possible; If the path leads to a node with an identifier, P is a keyword. If the path terminates before P, the string is not in the set of keywords. This takes clearly O (|P|) time.

The files that are to be outsourced are given to the trie. Each word is looked up in the trie to check whether it is a keyword and the number of occurrences is stored. This value is then passed on to the next phase, which is Indexing.

3.4 Indexing

Index is created as a list of mappings [10] which correspond to each keyword. The list for a particular keyword contains details such as:

1. File ids of the files which has the particular keyword
2. Term frequency for each file which denotes the number of times the keyword has occurred in the file. This measures the importance of the keyword in that file.
3. Length of each file
4. Relevance score for each file
5. Number of files that has the particular keyword

Data structures such as B+ trees can be used to store this data. Term frequency, length of the file, number of files for the keyword are used to calculate the relevance score for each file by scoring mechanisms which is discussed later in the Ranking modules.

The previous papers discuss architectures [5][6] where both the index and the files are stored in encrypted form in the un-trusted server. Whenever user searches for a word, the request is sent to the un-trusted server, which searches over the index and sends the entire mapping that is created for the word to the user. The user has the overhead to decrypt and request to retrieve the most relevant files based on the relevance score information in the index. This takes up a huge amount of bandwidth and round trip time. To reduce the overheads, a new architecture that stores the index as plain text in the local trusted server is proposed. When user searches for a word, the word is sent to the local trusted server, which searches the index, finds out the most relevant files and requests un-trusted server for the files to be retrieved and sent to user thereby ensuring data confidentiality in un-trusted server.

Whenever a data file is stored, it is preprocessed to generate a index containing the aforesaid details using the keywords extracted (using multiple string matching algorithm discussed earlier) from the data file. The index creation scheme is as follows:

```
1. For each wi, that belongs to the keyword set W, generate F(wi) which
   denotes the file ids that contain wi
2. For each wi € W
       For 1 <= j <= |F(wi)|
           2.1. Calculate score of the file Fij (with the help of
                scoring mechanisms discussed later) and store as Sij
           2.2. Store it with file id id(Fij), length of the file |Fij|
                as (id(Fij) || |Fij| || Sij) in I(wi) which is the index
                list for the particular word wi
           2.3. Update the total number of files that contain the
                keyword with the index list as (I(wi)||N)
```

3.5 Ranking

Once the documents are stored and indexed, the next important function is to rank them using details available such that the user retrieves the top ‘k’ most relevant documents. To do so, we need to calculate a numeric score for each file. In the IR community, the most widely used ranking functions are based on the TF X IDF rule, where TF stands for Term frequency which represents the number of times a keyword is present in a file and IDF stands for Inverse Document Frequency which is defined as the ratio of number of file containing the word to the total number of files present in the server.

The Ranking Function [5] used:

$$\text{Score }(W,Fi) = \sum 1/|Fi| \,.\, (1 + \ln fi,t) \,.\, (1 + N/ft)$$

W: Keyword whose score to be calculated
fi,t : Frequency of term in file Fi
|Fi|: Length of the file
N : Total number of files in the collection.

4 Conclusions

In this paper, we solve the problem of post processing overhead and unnecessary network traffic created when Boolean search techniques are used, by introducing the ranked keyword search scheme. The scheme generates indexes that help the user to search for his documents in a secure environment. The files matching the keyword search are further ranked based on the relevant score calculated with term frequency, file length etc.

Further extensions to the project can be done by

1. Supporting multi user environment where there would be an extra entity in the scenario i.e., data user who is authorized to access other users files. The authorization mechanisms and key exchanges methods can be modified to support the same.
2. Tolerating minor typos and format inconsistencies that occur while typing the key words. This can be done by introducing fuzzy keyword mechanism discussed earlier.

References

1. Song, D., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Proc. of IEEE Symposium on Security and Privacy 2000 (2000)
2. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
3. Chang, Y.-C., Mitzenmacher, M.: Privacy preserving keyword searches on remote encrypted data. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 442–455. Springer, Heidelberg (2005)
4. Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., Lou, W.: Fuzzy keyword search over encrypted data in cloud computing. In: Proc. of IEEE INFOCOM 2010 Mini-Conference (2010)
5. Wang, C., Cao, N., Ren, K., Lou, W.: Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data. IEEE Transactions on Parallel and Distributed Systems 23(8) (2012)

6. Wang, C., Cao, N., Li, J., Ren, K., Lou, W.: Secure ranked keyword search over encrypted cloud data. In: Proc. of ICDCS 2010 (2010)
7. Curtmola, R., Garay, J.A., Kamara, S., Ostrovsky, R.: Searchable symmetric encryption: improved definitions and efficient Constructions. In: Proc. of ACM CCS 2006(2006)
8. Rajan, R.: Efficient and Privacy Preserving Multi User Keyword Search For Cloud Storage Services. International Journal of Advanced Technology and Engineering Research (IJATER) 2(4) (2012) ISSN 2250 - 3536
9. Khan, Z.A., Pateriya, R.K.: Multiple Pattern String Matching Methodologies: A Comparative Analysis (2012)
10. Witten, I.H., Moffat, A., Bell, T.C.: Managing gigabytes: Compressing and indexing documents and images. Morgan Kaufmann Publishing, San Francisco (1999)
11. Moataz, T., Shifka, A.: Boolean Symmetric Searchable Encryption

Combining the X.509 and the SAML Federated Identity Management Systems

Marcus Hardt[1], Arsen Hayrapetyan[1], Paul Millar[2], and Shiraz Memon[3]

[1] Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Germany
firstname.lastname@kit.edu
[2] Deutsches Elektronen Synchrotron, Germany
p.millar@desy.de
[3] Jülich Supercomputing Centre, Germany
a.memon@fz-juelich.de

Abstract. Every distributed computing infrastructure requires authentication and authorisation infrastructures (AAI) to manage access to resources and content. Several of such so called AAI systems are in use within different groups of users. In the Large Scale Data Management and Analysis project we aim to support and bring together many user communities. We therefore need to harmonise the currently used AAI systems. The approach described is to translate between different authentication systems. We furthermore try to maintain the same trust level wherever possible, and to harmonise authorisation across the involved systems.

1 Introduction

Various scientific communites are developing new techniques and equipment for collecting data at increasing rates. The resulting data-deluge is challenging their ability to manage this data. Also, these communities are geographically distributed. Therefore, existing approaches to data management and access control seize to function well. One of the key questions faced by new approaches is that of how to handle authentication and authorisation in a federated environment. Traditional Authentication and Authorisation Infrastructures (AAI) are based around centrally managed user accounts and groups. This does not scale to the volume and flux of both users and data. Therefore, so called federated identity management approaches are required.

If each of the participating communities independently discovered their own approaches for handling access to its data, that would result in multiplied and therefore wasted efforts. The German Helmholtz Association has therefore funded the Large Scale Data Management and Analysis (LSDMA) portfolio extension, charged to stimulate and drive innovation within the academic sphere and targeted at improving the ways how data are stored, managed and analysed.

The LSDMA project gathers 25 different user communities grouped by their scientific domain. This covers use-cases from electro mobility, battery testing and simulation, climate modelling, human brain image analysis, selective plane

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 404–415, 2014.

microscopy, synchrotron tomography, high energy physics and more. Common requirements include performant, safe and simple access to data, capabilities to search within the data, archival and analysis of data.

As some of these user communities already have existing AAI systems, the LSDMA project does not attempt to change the ways users are working today. Instead, the goal is to provide a flexible service that can bridge between the existing AAI Infrastructures. Furthermore, many services support only one AAI system. Hence, the ability to bridge between technologies yields at making services that only support one AAI available to users from another AAI system. In addition to the technical challenges of authentication and authorisation, the level of trust needs be maintained wherever possible.

The relevant federated identity management technologies in this field are X.509 and SAML, both token based. Outside the scientific sector, OAuth and OpenID are widely used. While the focus of this paper is on the first two, the latter two can be integrated into the presented approach, because they are also token based. Plain username / password systems are not considered in this paper.

For the sake of clarity, this paper focuses on a hypothetical but practical use-case, which we will use for illustrating the presented solution details. This use case describes a web-portal that provides a visualisation service. The data for visualisation is stored on a third-party storage. The user authenticates to the web-portal with their SAML credentials while the storage service requires X.509 certificates for providing access to the data. Of course, both services could provide alternative authentication methods. However, similar requirements have emerged in the requirement analysis of the LSDMA project. To support this use-case, the authentication token of the user must be translated to one supported by the service. One notable impact of this design will be that distinct scientific communities will be able to access each others services without neither changing their existing tools nor services.

In order to be useful in practice, a solution that combines different AAIs, has to be able to fulfil the following requirements:

1. Hide its complexity from the user
2. Work well on commandline and in a browser
3. Support interactive and non-interactive authentication
4. Support delegation of data access rights to a web portal
5. Support generic delegation of data access rights to any host
6. Translate between the federated identity management tokens
7. Provide support for the mapping to the respective authorisation decisions

The remainder of this paper is organised as follows: In section 2 we give an overview of existing relevant technologies. In section 3 our approach for the integrative architecture is presented. Section 4 gives the conclusion and identifies the future work.

2 State of the Art

A variety of technologies exist that relate to federated authentication and authorisation standards. In this chapter we present those of them which are relevant to our work. The list, however, is not meant to be exhaustive.

2.1 X.509 Based Authentication and Authorisation

The X.509[14] based authentication and authorisation standard is widely used by scientific grid communities. Examples include Worldwide LHC Computing Grid and Open Science Grid. It is used, as well, by numerous web applications, for example, for secure online banking. In this section we are focusing on the grid communities, because they are more relevant to the LSDMA project.

The X.509 based authentication in the grid makes use of several components described briefly below.

The communicating end entities (EEs: users, hosts, services) in the grid, along with one or more Certification Authorities (CAs), are constituents of a Public Key Infrastructure (PKI). The CAs, also known as trusted third parties (TTP), provide EEs with digitally signed X.509 certificates containing their identity information. These certificates are used to authenticate the EEs. The grid CAs are established commonly on a per-country basis. The grid collaborations are geographically distributed over many institutes in different countries. Thus, the structure of a grid collaboration imposes the requirement of trust of the "foreign" CAs by all EEs. This problem is not scalable per se. To solve the pairwise-trust problem, the International Grid Trust Federation (IGTF)[17] has been established. This group maintains a set of minimum requirements for its member CAs (e.g., that a user's identity must be manually vetted using a passport or an equivalent government-issued document before issuing a user certificate). IGTF verifies routinely the CAs' conformance to the requirements. The resource sharing sites in the grid collaboration install the veirified IGTF CA certificate bundle on their resources, thus enabling mutual authentication between EEs on the collaboration scale.

Several CAs in IGTF provide Short-Lived Credential Services (SLCS) for their users. SLCS allows users to get X.509 certificates online without getting through the thorough identity vetting procedure (involving face-to-face meeting) for the standard X.509 certificates. To compensate this "lightweight" identity vetting procedure and improve the security, the SLCS certificates have much shorter maximum validity period than the standard ones (one million seconds versus 13 months). After the old SLCS certificate has expired users can easily request a new one. In Germany, the IGTF-accredited SLCS certificates are provided by DFN SLCS CA.

The X.509 based authorisation in the grid requires the authentication information of an EE and additional data about EE privileges to access shared resources of the collaboration. We describe the relevant components briefly below.

The scientific grid communities contribute their computational and storage resources to Virtual Organisations which are dedicated to specific scientific research. The EEs register with the VO and get assigned specific groups and roles according to the work they perform within their VO. The group and role information is used later by the authorisation services to grant or deny access to the resource. The VO membership, group and role information is managed by the VO via a dedicated central service.

Single sign-on and identity delegation are supported in grid VOs through proxy certificates. A proxy certificate [11] is digitally signed by a user delegating her identity for various grid tasks, for example, accessing files on a remote storage. For improved security each identity delegation is bound to certain public key. Furthermore, the proxy lifetime is limited (the default being 12 hours).

There are many *implementations* for components involved in X.509-based authentication and authorisation in the grid. We list few of them below.

Tools for managing X.509 certificates vary between command line (e.g. OpenSSL [4], gLite UI) and the web interfaces provided by CAs (e.g. German Grid CA web interface).

The SLCS CA front-end is provided by GridShib, a project based on the Shibboleth SP component. Software for CA management and an online interface for web-based certificate requests is usually custom-made by CA. Projects providing such software include OpenCA and EJBCA.

Two standards for grid VO membership management are VOMS[5] provided by gLite software and Unity (formerly known as UVOS [6]) provided by Unicore. Both support grouping of user accounts and assigning specific roles to them.

Proxy certificate management is provided by every grid middleware package. Examples include Globus, gLite, Unicore.

Despite the technical advantages of X.509, many users are uncomfortable with using certificate-based authentication. Reasons include an IGTF-requirement to update certificates annually, the lack of built-in web browser support for proxy certificates, and the fact that grid tools have distinct trust stores from web-browsers. Since many IGTF CAs use web portals for user interaction, this last item requires users to go through a convoluted export procedure before they may use their certificate for authentication in a grid context.

Regarding the requirements we have identified in section 1, X.509 does support [2, 3, 5]

2.2 SAML Based Authentication and Authorisation

The SAML based authentication and authorisation are used by numerous scientific and non-scientific communities. Examples include the AAI of the German Research Network (Deutsches Forschungsnetz, DFN [2]) and the AAI of the Swiss Research and Education Network (SWITCH).

The Security Assertion Markup Language (SAML) [8] was developed later than X.509. It is an XML-based open standard for specifying authentication and authorisation data. These data are expressed in the form of SAML assertions that are typically exchanged between Identity Provider (IdP) and Service Provider (SP).

Although there are several profiles for *SAML based authentication*, the most commonly used is the Web Browser Single Sign-On (Web-SSO) profile. In this profile a principal (i.e. a user) wishes to authenticate to a Service Provider (SP) using a web-browser. The SP makes use of web based redirects, so that the user can authenticate himself to the Identity Provider (IdP) of his home institution. After successful authentication, the IdP uses another web redirect, so that the web browser delivers the SAML assertion to the SP. These redirects allow passing information between IdP and SP via the users' browser, therefore no direct connection between IdP and SP is required for authentication.

The Web-SSO profile assumes the use of a web browser as a user agent which makes it not suitable for non-browser applications like desktop applications or server-side code running as a web application. A companion SAML profile known as Enhanced Client or Proxy (ECP) profile is available that removes the limitations of Web-SSO profile designed around limitations of the web browser.

To manage trust relationship between IdPs and SPs, SAML based federations can be formed. The federations set policies which the members of federations adhere to. They also vet their member IdPs and SPs and maintain a list of their members. The SPs can therefore rely on the federation policy to expect the minimum set of information being released by the IdPs to them. On the other hand, the IdPs can rely on federation policy to expect that the released data will be used appropriety. This simplifies the trust relationships, since instead of having multiple bilateral agreements with IdPs, the SPs can only have one agreement with the federation.

In many cases the national research network providers (e.g. DFN in Germany) operate such trust federations. Depending on the level of trust (e.g. the quality level of the user-ID-vetting, information expiry, etc.) different federations may be formed. In case of Germany there are three federations: DFN-advanced, DFN-basic and DFN-test, the first of which has comparable requirements as imposed by the IGTF policies in the X.509 domain.

To enable trustworthy authentication and authorisation, information exchange and to share resources on larger scale, SAML based federations can interfederate. When two federations interfederate, they agree to trust the credentials of each other's member IdPs and SPs. One of the most prominent intiatives in this area is eduGAIN project [9].

The *SAML based authorisation* by the SP is in most cases based on the authentication data provided by the IdP in the SAML assertion. In addition, the users from different IdPs can be grouped according to certain criteria. The group information can be taken into account by the SPs when making authorisation decisions.

SAML does support delegation, but not in the generic manner as X.509 does. It typically involves a lot of overhead, because all involved IdPs and SPs have to authorise delegation. One typical use-case, however, is an exception. This is the portal delegation, which may be used, in case a web-portal needs to access external information on behalf of the user.

SAML based authentication and authorisation solutions are *implemented* by serveral products. We list some of them below.

The most known SAML implementation is Shibboleth [3], whose developers claim they have the world's most widely deployed federated identity solution. There are also many other Open Source implementations. Examples include OpenSAML, simpleSAML-php, ZXID, Lasso and OpenSSO. SAML based group management is provided by a software called GMT, developed at SWITCH.

Although this paper focuses on SAML Web-SSO, it is worth mentioning that projects exist that aim to add SAML authentication to GSS- and SASL- authentication frameworks. Since many common Internet protocols support either GSS- or SASL- authentication, such approaches (if successful) will bring SAML-based authentication to the majority of non-web applications. There are currently two major approaches: Application Bridging for Federated Access Beyond Web (AB-FAB) [13] implemented by Project Moonshot and ECP-over-GSS [7].

In terms of the identified requirements SAML does support [1, 2, 4]

2.3 Credential Translation

Credential translation refers to a process of generating authentication or authorisation tokens or credentials for a given AAI service based on the authentication with other types of credentials. In this paper we are particularly interested in the token based credential translation from SAML assertions to X.509 certificates or proxies. The credential translation service can be a web service which requires SAML authentication and uses the resulting SAML assertion to request the X.509 certificate on users's behalf. The online CA issuing the X.509 certificate is an SP configured to accept SAML assertions delegated by the user to the credential tranlsation service. The certificates issued by the online CA can be short- or long-lived depending on the trust between the CA and the IdP authenticating the user.

The implementations of credential translation services from SAML to X.509 tokens include gridcertlib [15], a java library developed by SWITCH as well as Security Token Service (STS) developed within the EMI project. The SAML assertion delegation is supported by the GridShib [16] software which makes it suitable as a base for custom-developed credential translation web services.

IGTF-accredited SLCS certificates are issued by several online CAs based on SAML authentication. DFN SLCS CA and SWITCH SLCS CA issue short-lived X.509 certificates while the Terena credential service (TCS) issues long-lived ones.

2.4 Other Federated Authentication and Authorisation Technologies

There is a number of federated authentication and authorisation technologies and standards other than X.509 and SAML based standards. In this section we describe two of the most prominent and widely used ones.

OpenID is an open standard for federated authentication. It offers user-centric authentication mechanism allowing the user to choose the OpenID Provider (i.e. identity provider) for asserting her identity for the relying party (e.g. service provider). Most importantly, the standard does not require the trust relationship to be established between relying party and OpenID Provider in advance. The consumers can work with all possible OpenID providers. This makes it useful for the resource providers who are interested in offering users convenient and fast access to their resources. On the other hand, the relying party should not rely on the OpenID Provider for the trustfullness of the identity information about the user. Thus, the OpenID standard can be the choice in the cases when the trust requirement with respect to identity vetting and end-users' privacy are not the the primary concern of the AAI.

Many content management systems and web-based services provide plugins for OpenID support. It can be enabled also via libraries implemented in many languages.

There are several projects and standards aimed at improving the security level provided by the OpenID protocol. Examples include integration of the OpenID protocol into the SAML IdP (simpleSAMLphp) and integrating OpenID with OAuth (OpenID Connect standard, draft).

OAuth v2.0 is an open standard for authorisation. It allows third-party applications to get access to a web resource with the approval of the resource owner. In the most common OAuth scenario the service provider accepts a third-party Client application to access the data owned by the user based on an access token issued by the Authorisation Server. The access token contains the user's identity information which can be released to the authenticated Client if the user approves it. The user approves the release of her personal data by authenticating to the Authorisation Server. An important implication of this protocol is that the users never share their credentials with their Clients when delegating them the task of accessing their resources.

The implementations of OAuth components are available as libraries for various languages including Java and Python. There are standardisation efforts aimed at integrating OAuth authorisation standard with SAML 2.0 authentication and OpenID authentication (OpenID Connect draft).

2.5 Authorization and Group Management

The ability to define and manage groups is not directly associated with federated identity management systems. However, often the membership in a group is used by a service to make the authorisation decision. Therefore, authentication, group definition and authorisation are closely related.

The two largest computing middlewares globus/gLite and Unicore both provide support for Virtual Organisations or VOs [12]. These VOs are used to form groups of users on side and to allocate hardware resources for VOs on the other side.

VOMS Within WLCG, group membership is managed and asserted through one or more Virtual Organisation Membership Service (VOMS). This service maintains a database of users and their group membership. It also provides a web interface for administrators to add and remove users from groups. When generating the grid-proxy-credential, a user may request an attribute certificate[11] from one or more VOMS server. The supplied attribute certificates are embedded within the proxy certificate so that, when authenticating with a remote service, the remote service is able to extract the attribute certificate. Group membership is then discovered, provided the service trusts the VOMS server that issued the attribute certificate.

Unity SAML also describes how an SP may query an IdP directly, requesting assertions about some particular user. This allows the SP to gather additional assertions about the user from third-party IdPs after the user has delivered an assertion through Web Browser SSO; for example, to query the group membership of this user. Such third-party group-membership services are broadly similar to VOMS servers; however, in contrast to VOMS, SAML group-membership is asserted when the user authenticates with the SP. All the attributes may be employed by the SPs when making an authorisation decision.

The Unicore [10,1] grid middleware comes with the Unicore VO Service, UVOS, now being renamed to Unity. It implements the VOMS concept based on the SAML and XACML standards. For this Unity and Unicore make use of attribute aggregation. This is a SAML technique in which a Service Provider (SP) aggregates a users attributes by querying several attribute services – such as Unity.

Extensions include (among others) support for a hierarchical VO structure and a pluggable support for additional interfaces, so that VOMS style attribute certificates can be generated. Despite the extensibility of Unity, both VO systems are different in structure and not currently able to exchange group definitions among each other.

3 Design and Integrative Architecture

In this section we present the architecture of the AAI for the LSDMA project. The AAI is designed to support different authentication scenarios involving SAML and X.509 credentials. Standard software components of the existing implementations will be used for AAI components (see top of fig. 1). For SAML-based federations these are: Shibboleth Identity Provider (IdP) and Service Provider (SP). For X.509-based PKI these are: Certification Authority (CA) and Virtual Organisation Management Service (VOMS). The translation from SAML to X.509 credentials will be enabled by a component generating an asymmetric key pair and requesting an X.509 certificate from the CA upon user's successful authentication at the IdP. With respect to our initially described hypothetical use-case, three relatively simple authentication scenarios can easily be supported by standard AAI setups as described on Figure 2.

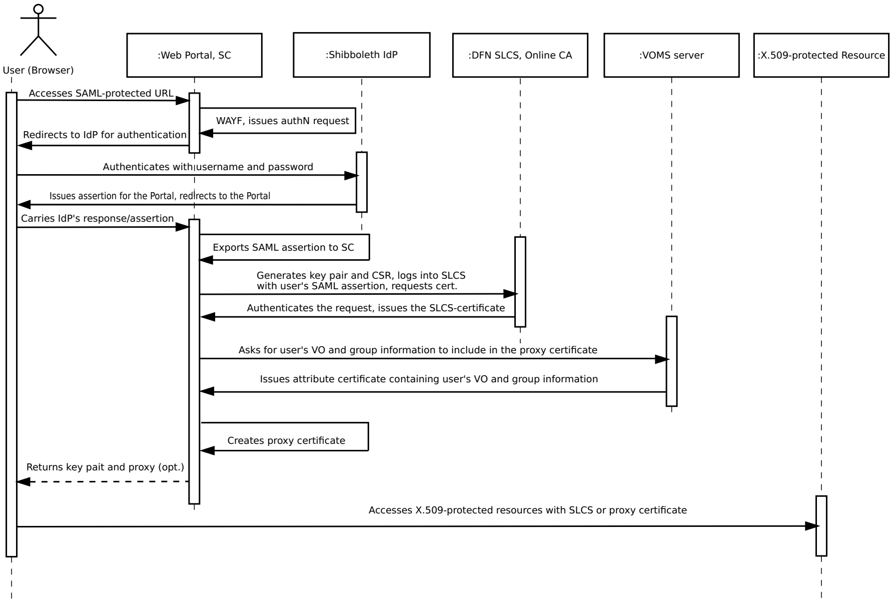

User (Browser)
:Web Portal, SC
:Shibboleth IdP
:DFN SLCS, Online CA
:VOMS server
:X.509-protected Resource
Accesses SAML-protected URL
WAYF, issues authN request
Redirects to IdP for authentication
Authenticates with username and password
Issues assertion for the Portal, redirects to the Portal
Carries IdP's response/assertion
Exports SAML assertion to SC
Generates key pair and CSR, logs into SLCS
with user's SAML assertion, requests cert.
Authenticates the request, issues the SLCS-certificate
Asks for user's VO and group information to include in the proxy certificate
Issues attribute certificate containing user's VO and group information
Creates proxy certificate
Returns key pait and proxy (opt.)
Accesses X.509-protected resources with SLCS or proxy certificate

Fig. 1. The components of the LSDMA AAI architecture

In the rest of this section we will focus on the complex case in which credential translation from SAML to X.509 formats is required. We will assume that the access to the visualisation data requires a valid proxy certificate, while the user authenticates with username and password at an IdP and uses the SAML Web-SSO profile. It is mportant to note that the user credentials have to be delegated to the Web portal running the visualisation program that uses the credentials to read and visualise the data. The corresponding sequence diagram is presented in Figure 1. In the following sections we describe the authentication process and components of the federated AAI in more detail.

3.1 SAML-Based Shibboleth Authentication

The user points her browser to the *Web portal* and requests data visualisation. The visualisation service at the web portal is protected by a *Shibboleth SP*. The SP then determines the user's home organisation and redirects her to the corresponding *IdP*. The IdP asks the user to authenticate, e.g. with username and password. Upon successful authentication the user returns to the portal carrying a valid SAML assertion.

3.2 SAML Assertion Delegation and SLCS Certificate Request

The SAML assertion is delegated by the portal to the *SLCS Client* (SC). At this point the SC generates an asymmetric key pair for the user as well as a

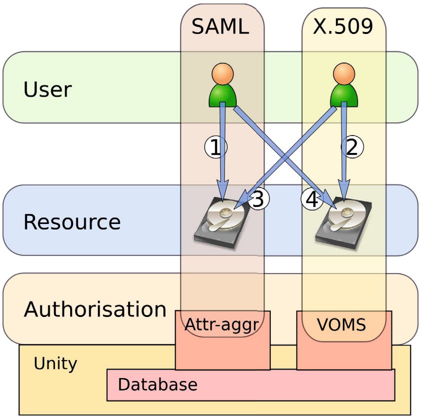


Fig. 2. The possible authentication scenarios in LSDMA. The simple ones are (1), where a SAML-authenticated user accesses a SAML-protected resouce and (2) where an X.509-authenticated users access an X.509 protected resouce. Credential translation is required, in cases (3) and (4). The first case (X.509 user accesses SAML resource) can be easily accomplished by allowing X.509 at the IdP. This cannot be centrally provided, because every single IdP has to allow this. Scenario (4), SAML user authenticates to X.509 resource is handled in this paper.

certificate signing request (CSR). The fields of the Distinguished Name (DN) of the user in the CSR, such as user name or organisation, are taken from the SAML assertion provided by IdP. The SC forwards the CSR to the *DFN SLCS* along with the user's SAML assertion for authentication. The DFN SLCS itself acts as Shibboleth SP which supports SAML assertion delegation and therefore can authenticate the delegated request. In particular, it verifies that the request is coming from a trusted delegate, our Web portal, and carries a valid SAML assertion from a trusted IdP. Upon successful authentication of the request DFN SLCS CA will issue the SLCS certificate for the user which is returned to the SC and stored on the portal. This may alreaaddy suffice for the user to access the data and complete his visualisation.

3.3 VOMS Proxy Generation

As part of a more complicated scenario the user and the storage hosting the data for visualisation may be part of a Virtual Organisation (VO). The later controls the access rights on the resources based on the role a user has in the VO. In our case the visualisation application must have a proxy certificate incorporating user VO membership information in order to get access to the data. To fulfill this requirement, the SC will contact the *VOMS server* for the VO to fetch user's VO memebership information and incorporate it into the proxy.

3.4 Putting It All Together

Once the proxy is generated, the visualisation application is able to access the data on the storage on user's behalf and produce visualisation objects and send them to the user's browser.

As it can be seen from the description above, the central component which is acting as a bridge between SAML-based and X.509 based authentication realms, is the SC. For its implementation we are currently considering two possibilities: i) adaptation of the GridCertLib java library, initially developed for SWITCHaai [18], to the existing DFN SLCS; ii) our own implementation of the component based on the example implementation provided by GridShib [16] project.

Regarding the requirements formulated in the introduction, the presented solution supports [1, 2, 3, 4, 5, 6, 7] However, (2) is only supported with IdP that support ECP and (3) supports automated authentication up to 10 days after the initial SAML login was carried out.

4 Conclusions and Future Work

The diverse user communities within the LSDMA project use federated identity management solutions from the two domains of SAML and X.509, both for authentication to services and for authorisation within the service.

In this paper we have presented an integrative architecture that is capable of translating authentication tokens between both domains. We have furthermore shown that group management for authorisation decisions can be supported, too. Currently the presented example of VOMS does however neither take existing attributes of the SAML assertion into account nor would it support group definition for a SAML SP. Future work will therefore include the choice of one group definition platform. This platform needs to contain all group definitions. It will then need to be extended so that interfaces to both, SAML (via attribute-aggregation) and to X.509 (via VOMS).

One option for this is to use Unity for group definition and to extend it so that VOMS proxy certificates can be issued. In this way, SAML and X.509 secured services can base their authorisation decisions on an identical group definition. Furthermore, we foresee to explore the hierarchical group definition concept of Unity to facilitate the creation of subgroups to facilitate data sharing.

Additional technologies, predominantly used outside the scientific sector include OpenID and OAuth. The presented architecture was designed with the goal of being able to include both. The envisaged use cases include authentication to our SC using OpenID (to translate from OpenID to X.509) as well as including OAuth, so that users who have authenticated to our SC, using either SAML, X.509 or OpenID, can subsequently issue authorisations using OAuth. However, work on this is still in a very preliminary state and will be pursued in the future.

Some of the methods we describe are targeted at web-browser based activities. While web browsers are often useful for many workflows, this may not the case when working with specialised clients, such as commandline clients. With the advent of the SAML profile ECP, also commandline access will be available for SAML users.

Acknowledgement. This work is supported by the Portfolio Extension of the German Helmholtz Association "Large Scale Data Management and Analysis" [19].

References

1. Unicore summit (2012), http://hdl.handle.net/2128/4705 (last visited August 26, 2013)
2. DFN. The German National Research Network Provider, http://dfn.de (last visited June 1, 2013)
3. Shibboleth. Project homepage, http://shibboleth.net
4. The OpenSSL Team. OpenSSL project homepage, https://www.openssl.org/ (last visited October 10, 2012)
5. Alfieri, R., Cecchini, R.L., Ciaschini, V., dell'Agnello, L., Frohner, A., Gianoli, A., Lõrentey, K., Spataro, F.: VOMS, an authorization system for virtual organizations. In: Fernández Rivera, F., Bubak, M., Gómez Tato, A., Doallo, R. (eds.) Across Grids 2003. LNCS, vol. 2970, pp. 33–40. Springer, Heidelberg (2004)
6. Benedyczak, K., Biala, P.: Next generation of virtual organizations in unicore. In: Unicore Summit 2012 Proceedings (2012)
7. Cantor, S., Josefsson, S.: SAML Enhanced Client SASL and GSS-API Mechanisms. IETF Draft Document (2013), https://datatracker.ietf.org/doc/draft-cantor-ietf-kitten-saml-ec/ (last visited November 13, 2013)
8. Cantor, S., Kemp, J., Philpott, R., Maler, E.: Assertions and protocols for the oasis security assertion markup language (SAML) v2.0 (2005)
9. eduGAIN. Project homepage, http://edugain.org
10. Erwin, D., Snelling, D.: UNICORE: a grid computing environment. In: Euro-Par 2001 Parallel Processing, pp. 825–834 (2001)
11. Farrell, S., Housley, R.: RFC 3281: An internet attribute certificate profile for authorization. IETF RFC, http://www.ietf.org/rfc/rfc3281.txt
12. Foster, I.: The anatomy of the grid: Enabling scalable virtual organizations. In: Sakellariou, R., Keane, J.A., Gurd, J.R., Freeman, L. (eds.) Euro-Par 2001. LNCS, vol. 2150, pp. 1–4. Springer, Heidelberg (2001)
13. Howlett, J., Hartman, S.: Application Bridging for Federated Access Beyond web (ABFAB). IETF Draft, http://datatracker.ietf.org/wg/abfab/
14. ITU-T Study Group 17: Security. In: Public-key and attribute certificate frameworks (October 2010), http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=X.509 (last visisted August 22, 2013)
15. Murri, R., Maffioletti, S., Kunszt, P., Tschopp, V.: Gridcertlib: a single sign-on solution for grid web applications and portals, http://arxiv.org/abs/1101.4116v3
16. The GridShib Project. Homepage, http://gridshib.globus.org (last visited August 26, 2013)
17. The International Grid Trust Federation, http://www.igtf.net (last visited June 12, 2013)
18. The Switch AAI. Homepage, http://www.switch.ch/aai/ (last visited August 26, 2013)
19. van Wezel, J., Streit, A., Jung, C., Stotzka, R., Halstenberg, S., Rigoll, F., Garcia, A., Heiss, A., Schwarz, K., Gasthuber, M., Giesler, A.: Data life cycle labs, a new concept to support data-intensive science. arXiv e-print 1212.5596 (December 2012)

Analysis of Electronic Voting Protocol Using Strand Space Model

Venkatasamy Sureshkumar and Ramalingam Anitha

Department of Applied Mathematics and Computational Sciences,
PSG College of Technology, Coimbatore-641004, India
sand@mca.psgtech.ac.in, anitha_nadarajan@mail.psgtech.ac.in

Abstract. In this paper, we studied the electronic voting protocol to formalize and verify its fairness, privacy type properties in the strand space model. Also we established a suitable attacker model, the concept of pair bundles and equivalence of pair bundles which are used to frame the formalization and verification of fairness, privacy type properties in detail using the strand space model. For example, FOO protocol is considered to illustrate the model developed using the strand space theory. The result shows that the fairness, vote privacy properties are satisfied and receipt freeness property is failed for FOO protocol. Finally, an improvement of the FOO protocol is proposed so that it achieves receipt freeness property.

Keywords: Strand space model, e-voting protocol, pair bundle, equivalence of pair bundles.

1 Introduction

Electronic voting (e-voting) is getting growing attention from governments, mass media, and the scientific community. E-voting may be a feasible solution to increase voter participation in governmental elections and provide an efficient method to do away with “lost ballots”. There is a need for a system, which makes it easy for all citizens to cast their vote confidentially and verify the election outcome while eliminating the threat of fake votes, miscounting and uncertainty. Electronic voting protocol is useful in recording and tallying votes conveniently, efficiently and securely. Since the voting protocol is designed in such a way that it has to be convenient for manipulation, its properties become vulnerable to some attacks. In the case of paper based voting system, even when there is a flaw in the protocol there are some procedure to deduct fraud, such as votes counting in public, monitoring properly the ballot boxes at the time of transportation etc. But such procedures cannot be implemented in e-voting. Hence there is a need for formal verification of e-voting protocol so that the people have the same comfort with e-voting system that they have with the paper based system.

E-voting protocol may satisfy many properties such as fairness, eligibility, vote-privacy, receipt freeness, coercion-resistance, accuracy, verifiability, no

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 416–427, 2014.

unauthorized proxy and so on. The three properties vote-privacy, receipt freeness and coercion-resistance are called privacy type properties.

- **Fairness:** No leakage of votes. That is early voted results could not influence the remaining voters before tally.
- **Eligibility:** Voter can cast his vote only once, that too only legitimate voter. Only the authorized voter is allowed to vote and thus preventing fraudulent votes from being counted in tallying stage.
- **No Unauthorized Proxy:** If a voter decides not to cast his/her ballot, no party can take advantage of this and cast a forged ballot.
- **Vote-Privacy:** The system does not disclose the way in which the voter has voted. That is no third party can find out how a particular voter has voted.
- **Receipt- freeness:** Voter must neither be able to obtain nor construct a receipt to prove his vote to a third party.
- **Coercion- resistance:** A voter cannot prove to the coercer that he/she has voted in a particular way even when coercer is allowed to communicate with the voter during the voting stage. Coercion-resistance is a stronger property as we give the coercer the ability to communicate interactively with the voter.
- **Accuracy:** A casted vote cannot be altered and an invalid vote is not counted.
- **Individual verifiability:** A voter can verify that his/her vote has been taken into account.
- **Universal verifiability:** The published total number of votes is equal to the sum of all the votes polled.

It is impossible to design a protocol which satisfies all the above mentioned properties [3]. To check the correctness of these properties for a specific protocol, there is a need for formal verification. Many protocols thought to be truthful for a number of years were found to have some blemishes by using formal verification techniques [7]. So it is essential to use formal verification techniques to check the correctness of voting protocol prior to implementing them. Security properties of voting protocols are affirmed in natural language, to carry out the verification task, these properties are presumed to be formalized.

Related Work: There have been only a few attempts in formal modeling and verification of electronic voting protocols in the literature. Kremer and Ryan [4] modelled the FOO protocol [6] using applied pi calculus and expressed fairness and receipt-freeness as an observational equivalence. Another similar work [5],by the same authors dealt with a stronger notion of receipt-freeness namely coercion-resistance, obtained its relationship with receipt-freeness and vote-privacy using pi calculus. In [8], a generic and uniform formalization is given to define the notion of receipt-freeness using epistemic logic and it is expressed using indistinguishability relations associated with anonymity. The receipt-freeness in terms of the knowledge of agents is modeled using formal logic in [2]. In this paper, we present a general framework for formal analysis of a class of properties such as fairness, vote-privacy and receipt freeness. Moreover, the emphasis on a decision procedure distinguishes our treatment from others. In addition, we have improved FOO protocol which satisfies the receipt freeness but fails to hold universal verifiability. This is justified in [3] by showing that simultaneous achievement of universal verifiability and receipt-freeness is impossible in general. However, to the best of our knowledge this is the first formalization and verification of the properties of e-voting protocol in strand space model.

This paper is organized as follows. Section 2, briefly describes FOO protocol which is an electronic voting protocol. Section 3 recalls some terminologies of strand space model, defines an attacker model, the concept of pair bundles and the equivalence of pair bundles. Section 4, presents the formalization of FOO protocol and its properties in strand space model. In Section 5, we use our formal framework to analyze the security of FOO protocol. Section 6 concludes our work and provides several interesting directions for future work.

2 Electronic Voting Protocols

In literature three different kinds of e-voting protocols are available namely voting based on Mix-nets ,homomorphic encryption and blind signatures. Each of these common techniques is the basis for several schemes. There is no technique that is unanimously better than the other techniques. The potency of each depends on the situation in which it has been applied. The following section describes an e-voting protocol which is based on blind signature scheme.

2.1 FOO Protocol

This protocol involves three phases with three agents namely voter V, administrator A and compiler C. The administrator verifies the identity of the voter and signs on vote blindly, provided the voter is eligible to vote; otherwise he rejects the voter's requisition. Compiler collects the votes from all the voters, tallies the votes and publishes the result. This protocol also uses some cryptographic primitives such as blind signatures and security bit-commitments. Description of the FOO protocol is given in three phases as follows.

Phase-I: In this, voter V contacts the administrator A with his valid identification and commitment of his vote. The administrator checks whether the voter is eligible to vote with his identification, the administrator signs on the commitment blindly if the voter is eligible and rejects if he is not eligible to vote. The commitment of the vote is protected from the administrator's knowledge by the blinding scheme.

$$V \to A : \{\sigma_V \left[\chi\left(\xi\left(v, r\right), b\right)\right], Id_V\}$$
$$A \to V : \{\sigma_A \left[\chi\left(\xi\left(v, r\right), b\right)\right]\}$$

where ξ-commitment scheme with random number r known only to the voter V, χ- blinding scheme with blinding factor b and σ- message extractable signature scheme. Voter chooses his vote v and a random number r to construct the commitment $\xi(v, r)$. This commitment is blinded by using the blinding scheme χ with the blinding factor b. Finally the voter signs on blinded message $\chi(\xi(v,r),b)$ using the signature scheme σ. At the end of this phase, V receives signed blinded message $\sigma_A[\chi(\xi(v,r),b)]$ from the administrator and un-blind it by using the blinding factor b so that the signature falls on the commitment.

Phase-II: This phase is the actual voting phase of the protocol.

$$V \to C : \{\sigma_A\left[\xi\left(v,r\right)\right]\} .$$

In this phase, V sends $\sigma_A[\xi(v,r)]$, A's signature on the commitment of V's vote, to the compiler C without disclosing his identity. Compiler tests the correctness of the signature of the administrator, if the test succeeds then enters $(l,\xi(v,r),\sigma_A[\xi(v,r)])$ as the l^{th} item into his list otherwise compiler rejects the vote as an invalid vote.

Phase-III: This phase starts only when phase-II is completely over. That is, all the voters have polled their votes or may be after a fixed deadline.

$$\begin{aligned} &C \to V : \{l_i,\xi(v_i,r_i),\sigma_A[\xi(v_i,r_i)]\} \\ &V \to C : \{l,r\} . \end{aligned}$$

C publishes the list $(l_i,\xi(v_i,r_i),\sigma_A[\xi(v_i,r_i)])$ of commitments he obtained, the list number and the administrator signed commitment which will be useful for the universal verifiability. Voter checks whether his commitment and signed commitment are in the list, if found then the corresponding list number l is noted. Using an anonymous channel voter sends the random number r along with the list number l in which his committed vote is stored. Making use of the received random number r, compiler opens the l^{th} ballot and publishes the vote v.

An informal analysis is given in [12]. Such kind of argument gives only informal proof about the correctness of this protocol. But when an automated system generates plenty of protocols, to check the correctness of those protocols, there is a need for an automatic system which requires a formal proof technique. Hence we go for a formal proof technique using strand space model. The strand space model is attracting people's attention in recent years [9,10], because strand space theory translates the description of protocol and security property into graphs. The maximal merit of strand space model is its succinctness.

3 Strand Space Model

Strand space model has been proposed as a formal method for verifying the security goals of cryptographic protocols. Strand space model [7] is a hand proof technique which is a potential tool used to verify the correctness of security protocols in various domains. Strand space theory provides a structure to determine what security goals a cryptographic protocol accomplishes.

3.1 Preliminaries

This section presents the basic terminologies and notations used in the strand space model [7] for modeling and analyzing the actions of the participating principals in the protocol.

Message Algebra: The collection of all possible messages that can be exchanged in the protocol execution is called message algebra which is closed under the process of encryption, concatenation, signing, blinding and commitment. It is denoted by $\mathscr{A}$.

Terms: The elements of $\mathscr{A}$ are called terms.

Strand: For a participant, strand is a sequence of message sends and receives. Transmission of a term t is represented as $+t$ and reception of term t is represented as $-t$. In other words, a strand is a linear structure, a sequence of one principal's message transmissions and receptions.

Strand Space: The collection of all possible strands for the various legitimate parties involved in the protocol together with penetrator strands is called strand space. It is denoted by Σ.

Trace: In the strand space Σ the trace of a strand is defined as list of incoming and outgoing messages in the same order in which they are exchanged.

Node: A node is a pair $< s, i >$, with the strand $s \in \Sigma$ and i an integer satisfying $1 \leq i \leq length(tr(s))$, where $length(tr(s))$ represents length of trace of the strand s . If s is a strand, $< s, i >$ is the i^{th} node on s.

Casual relations: The relation $n_1 \Rightarrow n_2$, holds between nodes n_1 and n_2 if $n_1 = < s, i >$ and $n_2 = < s, i+1 >$. The relation $n_1 \rightarrow n_2$, represents inter-strand communication; it means that $term(n_1) = +t$ and $term(n_2) = -t$. The two relations $\Rightarrow$ and $\rightarrow$ are called casual relations and they jointly impose a graph structure on the nodes of Σ. The vertices of this graph are the nodes, and the edges are the union of $\Rightarrow$ and $\rightarrow$. In a strand space, if $n_1 = < s, i >$ is a sending node and $term(n_1) = +t$ then $uns\_term(n_1) = t$.

Bundle: Let C be a set of edges, and let N_C be the set of nodes incident with any edge in C. C is called a bundle if:

- C is finite.
- If $n_1 \in N_C$ and $term(n_1)$ is negative then there is a unique n_2 such that $n_2 \rightarrow n_1 \in C$.
- If $n_1 \in N_C$ and $n_2 \Rightarrow n_1$ then $n_2 \Rightarrow n_1 \in C$.
- C is acyclic.

A bundle is an acyclic graph and it is a portion of the strand space which can be large enough to signify a single run of the entire protocol, nodes are related by casual relation, which is a partial ordered relation [7]. A bundle in a strand space can be represented by a sub-graph of the nodes and edges expressing causal dependencies of the nodes. To formalize and prove the correctness of fairness property, we use the concept of equivalent bundles. Further we define and use the concept of pair bundles and equivalence of pair bundles in sections 3 and 5 respectively to analyze the privacy type properties of FOO protocol.

3.2 Attacker Model

For the verification of e-voting protocol, we consider a passive attacker model. A passive attacker can only intercept and read the messages but cannot initiate, delete or modify the messages exchanged in the protocol execution. For modeling fairness property of e-voting protocol we extend the concept of equivalent bundles and reinterpret mapping [11] which were used to model entity anonymity. Also we define the atomic message set and the attacker knowledge set suitable to our model.

Definition 1 (Atomic Message Set). *The atomic message set* (AMS) *is a set that contains all the atomic messages already known to the attacker, such as all possible principals name* (P) *who can be involved in the protocol execution, set of possible keys* (K) *known to the attacker which could be used for constructing cryptographic primitives in the protocol and set of possible atomic messages* (M) *such as nonce, which could be used for the protocol execution. Hence* $AMS = P \cup K \cup M$.

Definition 2 (Attacker Knowledge Set). *The attacker knowledge set*(AKS) *containing messages generated from messages of* AMS, *and by the attacker knowledge as follows:*

- *if* $x \in AMS$, *then* $x \in AKS$,
- *if* $m \in AKS$ *and* $k \in K$, *then* $\{m\}_k \in AKS$, *where* $\{m\}_k$ *is the encryption of* m *using the key* k,
- *if* $\{m\}_k \in AKS$ *and* $k^{-1} \in K$, *then* $m \in AKS$,
- *if* $x \in AKS$ *and* $k^{-1} \in K$, *then* $\sigma(x, k^{-1}) \in AKS$,
- *if* $\sigma(x, k^{-1}) \in AKS$ *and* $k \in K$ *then* $x \in AKS$,
- *if* $x \in AKS$ *and* $b \in AKS$ *then* $\chi(x, b) \in AKS$,
- *if* $\chi(x, b) \in AKS$ *and* $b \in AKS$ *then* $x \in AKS$,
- *if* $x \in AKS$ *and* $r \in AKS$ *then* $\xi(x, r) \in AKS$,
- *if* $\xi(x, r) \in AKS$ *and* $r \in AKS$ *then* $x \in AKS$.

Definition 3 (Reinterpret mapping)

A binary relation π *on* $\mathscr{A}$ *is a reinterpret mapping for* m *if*

$$\pi(m) = \begin{cases} m & if \quad m \in AKS \\ \alpha & if \quad m \notin AKS \end{cases}$$

where α *is a constant message unknown to the attacker.*

This definition can also be extended for strands. Suppose C is a bundle of the strand space Σ and $s \in \Sigma$ is a strand for an arbitrary principal, then π is a reinterpret mapping for s if

$$\pi(term(\langle s,\ i\rangle)) = \begin{cases} term(\langle s,\ i\rangle) \ if \quad term(\langle s,\ i\rangle) \in AKS \\ \alpha \qquad\qquad\quad if \quad term(\langle s,\ i\rangle) \notin AKS, \end{cases} 1 \le i \le length(tr(s)).$$

Definition 4 (Equivalent strands [11]). *Two strands s_1 and s_2 are equivalent strands if $\pi(s_1) = \pi(s_2)$, and it is denoted as $s_1 \cong s_2$. In this case, the number of nodes in every strand is equal, the signs of corresponding nodes are same and the corresponding items are same after reinterpret mapping.*

Definition 5 (Equivalent bundles [11]). *If two bundles C_1 and C_2 both have a strand to one entity and the two strands are equivalent strands, then these two bundles are equivalent bundles to the entity, and it is denoted as $C_1 \cong C_2$.*

Definition 6 (Pair-bundle). *Two bundles C_1 and C_2 are called a pair-bundle if C_1 has a strand s_1 to an entity A in the execution of a protocol P and C_2 has a strand s_2 to an entity B different from A but who plays the same role of A in the execution of the same protocol P. A pair-bundle is denoted by $\mathscr{B} = (C_1,\ C_2)$ in which C_1 and C_2 are called component bundles of $\mathscr{B}$.*

Definition 7 (Equivalent pair-bundles). *The two pair-bundles $\mathscr{B}_1$ and $\mathscr{B}_2$ are said to be equivalent pair-bundles if their corresponding component bundles are equivalent, and it is denoted as $\mathscr{B}_1 \cong \mathscr{B}_2$.*

4 Modeling Fairness and Vote Privacy Properties of FOO Protocol in Strand Space

In this section, we model FOO protocol as well as fairness and vote privacy properties of the protocol using strand space.

FOO protocol is modeled with voter's strand s_V, administrator's strand s_A and the compiler's strand s_C as below.

(i) Strand of the voter $s_V \in Voter[v, r, l, Id_V, \xi(v, r)]$.

According to the protocol provisions, its trace takes the following form:

$$\langle + \{\sigma_V[\chi(\xi(v,r),b], Id_V\},\ -\{\sigma_A[\chi(\xi(v,r),b]\},\ +\{\sigma_A(\xi(v,r)\},\\ -\{l, \xi(v,r), \sigma_A[\xi(v,r)]\},\ +\{l,r\}\rangle.$$

(ii) Strand of the administrator $s_A \in Admin[Id_V]$ with its trace of the form:

$$\langle - \{\sigma_V(\chi(\xi(v,r),b), Id_V\},\ +\{\sigma_A(\chi(\xi(v,r),b)\}\rangle.$$

(iii) Strand of the compiler $s_C \in Comp[r, l, \xi(v,r)]$ and its trace takes the form:

$$\langle - \{\sigma_A(\xi(v,r)\},\ +\{l, \xi(v,r), \sigma_A[\xi(v,r)]\},\ -\{l,r\}\rangle.$$

Definition 8 (FOO Space). *A FOO Space Ω is a strand space which has three kinds of strands administrator strands, voter strands and compiler strands.*

One run of a FOO protocol explained in section 2 can be represented as a directed graph by a bundle in strand space model as shown in Figure 1.

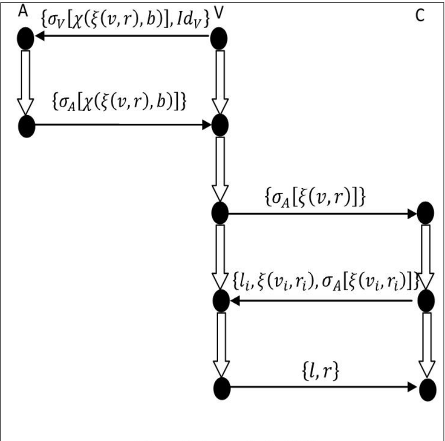

A
$\{\sigma_V[\chi(\xi(v,r),b)],Id_V\}$
V
C
$\{\sigma_A[\chi(\xi(v,r),b)]\}$
$\{\sigma_A[\xi(v,r)]\}$
$\{l_i,\xi(v_i,r_i),\sigma_A[\xi(v_i,r_i)]\}$
$\{l,r\}$

Figure1. A bundle in FOO space

Now we model fairness and vote privacy property of the protocol as below: An e-voting protocol is said to be fair if the two events one in which "principal V votes for a contestant 'a' " and another in which "principal V votes for the contestant 'b' " are indistinguishable [4]. Let s_{V_x} denote a strand for the voter V when he votes for the contestant 'x'. To prove that FOO protocol satisfies fairness property, it is equivalent to prove that $C_1 \cong C_2$, where $C_1,\ C_2$ are two arbitrary bundles with voter strands s_{V_a} and s_{V_b} respectively in FOO space $\mathit{\Omega}$.

An e-voting protocol is said to satisfy vote privacy if the two events one in which "principal V_1 votes for a contestant 'a', principal V_2 votes for a contestant 'b'" and another in which "principal V_1 votes for the contestant 'b', V_2 votes for the contestant 'a'" are indistinguishable [4]. To prove that FOO protocol satisfies vote privacy property it is enough to prove that $\mathscr{B}_1 \cong \mathscr{B}_2$, where $\mathscr{B}_1,\ \mathscr{B}_2$ are two arbitrary pair-bundles with the component bundles $C_1,\ C_2$ of $\mathscr{B}_1$ and $C_3,\ C_4$ of $\mathscr{B}_2$, in which C_1 has the voter strand $s_{V_{1_a}}$, C_2 has the strand $s_{V_{2_b}}$, C_3 has the voter strand $s_{V_{1_b}}$, and C_4 has the strand $s_{V_{2_a}}$.

5 Analysis of FOO Protocol in Strand Space

Detailed FOO protocol explained in section 2, modeled in section 4 is to be verified in this section using strand space model.

Proposition: 5.1. FOO protocol respects fairness, i.e., two arbitrary bundles C_1 and C_2 corresponding to a voter V in the FOO space are equivalent.

Proof. Let $\mathit{\Omega}$ be the FOO space and V be an arbitrary voting principal. The trace of 'V' takes the form

$$\langle +\{\sigma_V[\chi(\xi(v,r),b],Id_V\},\ -\{\sigma_A[\chi(\xi(v,r),b]\},\ +\{\sigma_A(\xi(v,r)\},\\ -\{l,\xi(v,r),\sigma_A[\xi(v,r)]\},\ +\{l,r\}\rangle.$$

Let s_{V_x} be the strand when V votes for the contestant x and s_{V_y} be the strand when he votes for a contestant $y \neq x$. Let C_1 and C_2 be two bundles in the FOO space such that s_{V_x} occurs in C_1 and s_{V_y} occurs in C_2. Since in phase-I and phase-II, vote casting process will be over, fairness property will not be affected by phase-III. So we consider bundles for the first two phases of the protocol.

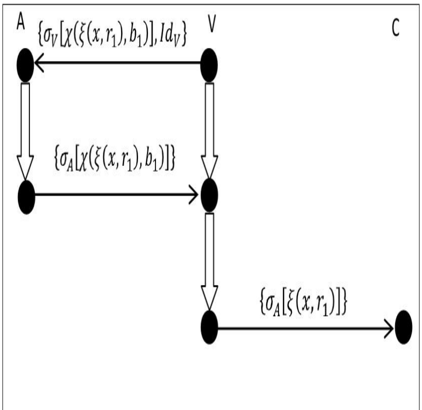

A
$\{\sigma_V[\chi(\xi(x,r_1),b_1)],Id_V\}$
V
C
$\{\sigma_A[\chi(\xi(x,r_1),b_1)]\}$
$\{\sigma_A[\xi(x,r_1)]\}$

Figure 2. FOO bundle when V votes for x

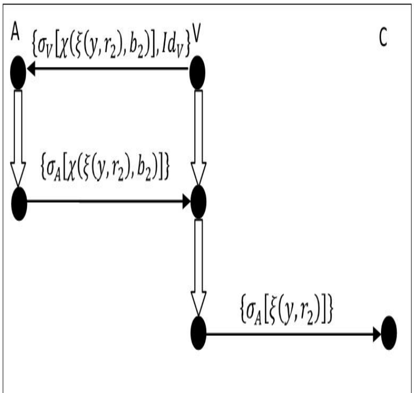

A
$\{\sigma_V[\chi(\xi(y,r_2),b_2)],Id_V\}$
V
C
$\{\sigma_A[\chi(\xi(y,r_2),b_2)]\}$
$\{\sigma_A[\xi(y,r_2)]\}$

Figure 3. FOO bundle when V votes for $y \neq x$

From the definition of equivalence of bundles, to prove that the two bundles C_1 and C_2 are equivalent, it is sufficient to prove that the two strands s_{V_x} and s_{V_y} are equivalent. Figure 2 and Figure 3 show that the numbers of nodes in both strands are equal; the signs of corresponding nodes are the same. Since $r_1,\ b_1$ and $r_2,\ b_2$ are in s_{V_x} and s_{V_y} respectively and kept secret by the voter, they do not belong to AKS and hence the corresponding items must be the same after reinterpret mapping. Therefore $\pi(s_{V_x}) = \alpha$ and $\pi(s_{V_y}) = \alpha$. Hence both s_{V_x} and s_{V_y} are equivalent. □

In the passive attacker point of view, the strands are indistinguishable and hence they are equivalent. This is true even when the administrator is corrupted in the sense that administrator's secret key is given to the attacker. Hence this fairness is a stronger one.

Proposition: 5.2 FOO protocol preserves vote privacy property, i.e., two arbitrary pair bundles corresponding to any pair of voters $(V_1,\ V_2)$ in the FOO space are equivalent.

Proof. Let $\mathscr{B}_1 = (C_1,\ C_2), \mathscr{B}_2 = (C_3,\ C_4)$ be two pair bundles in FOO space corresponding to the arbitrary voting principals pair $(V_1,\ V_2)$. Let $s_{{V_1}_{v_1}}$ be the strand when V_1 votes for a contestant v_1, $s_{{V_2}_{v_2}}$ be the strand when V_2 votes for a contestant v_2, $s_{{V_1}_{v_2}}$ be the strand when V_1 votes for the contestant v_2 and $s_{{V_2}_{v_1}}$ be the strand when V_2 votes for the contestant v_1 respectively. Assume that C_1, C_2, C_3 and C_4 are the bundles in FOO's space such that $s_{{V_1}_{v_1}}$ occurs in C_1, $s_{{V_2}_{v_2}}$ occurs in C_2, $s_{{V_1}_{v_2}}$ occurs in C_3 and $s_{{V_2}_{v_1}}$ occurs in C_4.

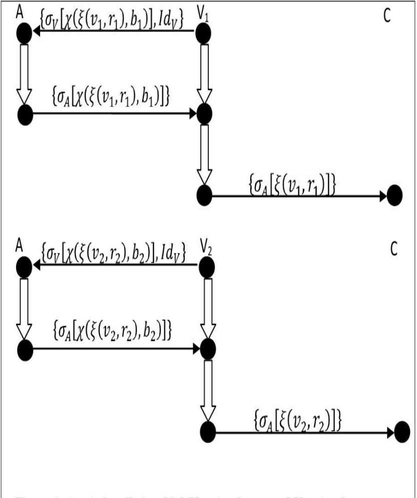

A
$\{\sigma_V[\chi(\xi(v_1,r_1),b_1)],Id_V\}$
V_1
C
$\{\sigma_A[\chi(\xi(v_1,r_1),b_1)]\}$
$\{\sigma_A[\xi(v_1,r_1)]\}$
A
$\{\sigma_V[\chi(\xi(v_2,r_2),b_2)],Id_V\}$
V_2
C
$\{\sigma_A[\chi(\xi(v_2,r_2),b_2)]\}$
$\{\sigma_A[\xi(v_2,r_2)]\}$

Figure 4. A pair bundle in which V_1 votes for v_1 and V_2 votes for v_2

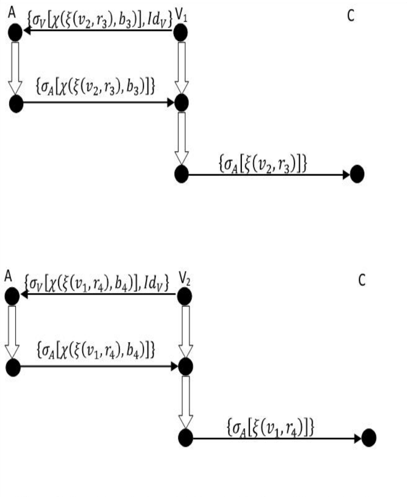

A
$\{\sigma_V[\chi(\xi(v_2,r_3),b_3)],Id_V\}$
V_1
C
$\{\sigma_A[\chi(\xi(v_2,r_3),b_3)]\}$
$\{\sigma_A[\xi(v_2,r_3)]\}$
A
$\{\sigma_V[\chi(\xi(v_1,r_4),b_4)],Id_V\}$
V_2
C
$\{\sigma_A[\chi(\xi(v_1,r_4),b_4)]\}$
$\{\sigma_A[\xi(v_1,r_4)]\}$

Figure 5. A pair bundle in which V_1 votes for v_2 and V_2 votes for v_1

From the definition of equivalence of pair bundles, to prove that the two pair bundles $\mathscr{B}_1, \mathscr{B}_2$ are equivalent it is sufficient to prove the equivalences of the component bundles $C_1 \cong C_3$ and $C_2 \cong C_4$. Figure 4 and Figure 5 show that the numbers of nodes in all the strands are equal; the signs of corresponding nodes are the same. Since $r_1,\ b_1$ in $s_{V_{1_{v_1}}}$, $r_2,\ b_2$ in $s_{V_{2_{v_2}}}$, $r_3,\ b_3$ in $s_{V_{1_{v_2}}}$ and $r_4,\ b_4$ in $s_{V_{2_{v_1}}}$ are kept secret by the voters, they do not belong to the AKS. Therefore $\pi(s_{V_{1_{v_1}}}) = \langle \alpha, \alpha, \alpha \rangle$, $\pi(s_{V_{2_{v_2}}}) = \langle \alpha, \alpha, \alpha \rangle$, $\pi(s_{V_{1_{v_2}}}) = \langle \alpha, \alpha, \alpha \rangle$ and $\pi(s_{V_{2_{v_1}}}) = \langle \alpha, \alpha, \alpha \rangle$. This shows the equivalences $s_{V_{1_{v_1}}} \cong s_{V_{1_{v_2}}}$ and $s_{V_{2_{v_2}}} \cong s_{V_{2_{v_1}}}$ and hence $C_1 \cong C_3$ and $C_2 \cong C_4$. □

Proposition: 5.3 FOO protocol does not satisfy receipt freeness property.

Proof. Construct the pair bundles $\mathscr{B}_1, \mathscr{B}_2$, the bundles $C_1,\ C_2,\ C_3,\ C_4$ and the strands $s_{V_{1_{v_1}}}, s_{V_{2_{v_2}}}, s_{V_{1_{v_2}}}, s_{V_{2_{v_1}}}$ as described in proposition 5.2. We give a counter example to show that the two pair bundles $\mathscr{B}_1,\ \mathscr{B}_2$ are not equivalent. Figure 4 and Figure 5 show the pair bundles $\mathscr{B}_1$ and $\mathscr{B}_2$ respectively. It is apparent that after reinterpret mapping $s_{V_{1_{v_1}}},\ s_{V_{1_{v_2}}}$ are not equal and $s_{V_{2_{v_2}}},\ s_{V_{2_{v_1}}}$ are not equal because when $r_1,\ r_2$ and $r_3,\ r_4$ are disclosed by the voters V_1 and V_2 respectively, $\pi(s_{V_{1_{v_1}}}) = \langle \alpha, \alpha, \sigma_A[\xi(v_1, r_1)] \rangle$, $\pi(s_{V_{2_{v_2}}}) = \langle \alpha, \alpha, \sigma_A[\xi(v_2, r_2)] \rangle$, $\pi(s_{V_{1_{v_2}}}) = \langle \alpha, \alpha, \sigma_A[\xi(v_2, r_3)] \rangle$ and $\pi(s_{V_{2_{v_1}}}) = \langle \alpha, \alpha, \sigma_A[\xi(v_1, r_4)] \rangle$. The attacker could distinguish the two pair nodes $< s_{V_{1_{v_1}}}, 3 >$, $< s_{V_{1_{v_2}}}, 3 >$ and $< s_{V_{2_{v_2}}}, 3 >$, $< s_{V_{2_{v_1}}}, 3 >$ represented in Figure 4 and Figure 5. Thus the strands $s_{V_{1_{v_1}}},\ s_{V_{1_{v_2}}}$ are not equivalent and $s_{V_{2_{v_2}}},\ s_{V_{2_{v_1}}}$ are not equivalent. Therefore $C_1,\ C_3$ are not equivalent and $C_2,\ C_4$ are not equivalent. Hence FOO protocol does not satisfy receipt freeness property. □

Failure of receipt freeness property implies the failure of coercion resistance property [1]. Hence FOO protocol does not satisfy coercion resistance property.

5.1 Improvement of FOO Protocol

Since FOO protocol does not satisfy receipt freeness property in voting phase, we can improve the protocol as follows:

Phase-1	$V \rightarrow A : \{\sigma_V \left[\chi\left(\{\xi\left(v,r\right),R\},b\right)\right],Id_V\}$ $A \rightarrow V : \{\sigma_A \left[\chi\left(\{\xi\left(v,r\right),R\},b\right)\right]\}$
Phase-2	$V \rightarrow C : \{\sigma_A \left[\{\xi\left(v,r\right),R\}\right]\}_{K_C}$
Phase-3	$C \rightarrow V : \{l_i, R_i\}$ $V \rightarrow C : \{l, r\}$

Phase-I: In this phase, V contacts A with his valid ID, blinded commitment which includes a partial secret R. A checks the validity of the voter and signs on the commitment blindly if V is an eligible voter.

Phase-II: The actual voting phase of the protocol is modified so that the correspondence of partial secret and the commitment is being confidential from the attacker. For that purpose we encrypt the entire message exchanged in this phase using the compilers public key K_C. V sends his commitment with A's signature, to the compiler C without disclosing his identity. Compiler test the correctness of the signature of the administrator, if the test succeeds then enter $(l, \xi(v,r), R)$ as the l^{th} item into his list otherwise compiler rejects the vote as an invalid one.

Phase-III: In this phase, C publishes the list (l_i, R_i) consisting of the list number and the corresponding partial secret which will be useful for the individual verifiability. Voter checks whether his partial secret is in the list, if found then the corresponding list number l is noted. Using an anonymous channel voter sends the random number r along with the list number l in which his committed vote is stored. Making use of the received random number r, compiler opens the l^{th} ballot and publishes the vote v.

Proposition: 5.4 Improved FOO protocol satisfy receipt freeness property.

Proof. Construct the strands $s_{V_{1_{v_1}}}$, $s_{V_{2_{v_2}}}$, $s_{V_{1_{v_2}}}$ and $s_{V_{2_{v_1}}}$ as described in proposition 5.2 whose traces are of the form

$$\langle + \left\{\sigma_V \left[\chi\left(\{\xi\left(v_1,r_1\right),R_1\}_{K_C},b_1\right)\right],Id_{V_1}\right\}, - \left\{\sigma_A \left[\chi\left(\{\xi\left(v_1,r_1\right),R_1\}_{K_C},b_1\right)\right]\right\}, + \left\{\sigma_A \left[\{\xi\left(v_1,r_1\right),R_1\}_{K_C}\right]\right\}\rangle,$$

$$\langle + \left\{\sigma_V \left[\chi\left(\{\xi\left(v_2,r_2\right),R_2\}_{K_C},b_2\right)\right],Id_{V_2}\right\}, - \left\{\sigma_A \left[\chi\left(\{\xi\left(v_2,r_2\right),R_2\}_{K_C},b_2\right)\right]\right\}, + \left\{\sigma_A \left[\{\xi\left(v_2,r_2\right),R_2\}_{K_C}\right]\right\}\rangle,$$

$$\langle + \left\{\sigma_V \left[\chi\left(\{\xi\left(v_2,r_3\right),R_3\}_{K_C},b_3\right)\right],Id_{V_1}\right\}, - \left\{\sigma_A \left[\chi\left(\{\xi\left(v_2,r_3\right),R_3\}_{K_C},b_3\right)\right]\right\}, + \left\{\sigma_A \left[\{\xi\left(v_2,r_3\right),R_3\}_{K_C}\right]\right\}\rangle,$$

$$\langle + \left\{\sigma_V \left[\chi\left(\{\xi\left(v_1,r_4\right),R_4\}_{K_C},b_4\right)\right],Id_{V_2}\right\}, - \left\{\sigma_A \left[\chi\left(\{\xi\left(v_1,r_4\right),R_4\}_{K_C},b_4\right)\right]\right\}, + \left\{\sigma_A \left[\{\xi\left(v_1,r_4\right),R_4\}_{K_C}\right]\right\}\rangle$$

respectively. since K_C^{-1} is kept secret, it is clear that $\pi(s_{V_{1_{v_1}}}) = \pi(s_{V_{1_{v_2}}})$ and $\pi(s_{V_{2_{v_1}}}) = \pi(s_{V_{2_{v_2}}})$ which implies that $C_1 \cong C_3$ and $C_2 \cong C_4$ and hence $\mathscr{B}_1 \cong \mathscr{B}_2$.

6 Conclusion

In this paper, we have discussed the formal verification of the FOO protocol and its properties. Using the notion of equivalent bundles in strand space model, fairness property is formally defined and hence its correctness is proved against passive attacker model. Extension of equivalent bundle concept for pair bundles is used to verify the correctness of privacy type properties. The vote-privacy property is proved to be correct and also we have obtained a counter example which illustrates that the failure of receipt freeness property. For future work this model is to be extended for active attacker and FOO protocol is to be fixed for coercion resistance property.

References

1. Backes, M., Hritcu, C., Maffei, M.: Automated verification of remote electronic voting protocols in the applied pi-calculus. In: 21st IEEE Comp. Secur. Foundations Symposium, CSF 2008, pp. 195–209 (2008)
2. Baskar, A., Ramanujam, R., Suresh, S.P.: Knowledge-based modelling of voting protocols. In: Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge, pp. 62–71. ACM (2007)
3. Chevallier-Mames, B., Fouque, P.-A., Pointcheval, D., Stern, J., Traoré, J.: On some incompatible properties of voting schemes. In: Chaum, D., Jakobsson, M., Rivest, R.L., Ryan, P.Y.A., Benaloh, J., Kutylowski, M., Adida, B. (eds.) Towards Trustworthy Elections. LNCS, vol. 6000, pp. 191–199. Springer, Heidelberg (2010)
4. Delaune, S., Kremer, S., Ryan, M.: Verifying privacy-type properties of electronic voting protocols. J. Comp. Secur. 17(4), 435–487 (2009)
5. Delaune, S., Kremer, S., Ryan, M.: Coercion-resistance and receipt-freeness in electronic voting. In: Proceedings of the 19th IEEE Comp. Secur. Foundations Workshop, pp. 28–39. IEEE Comp. Soc. Press (2006)
6. Fujioka, A., Okamoto, T., Ohta, K.: A Practical Secret Voting Scheme for Large Scale Elections. In: Zheng, Y., Seberry, J. (eds.) AUSCRYPT 1992. LNCS, vol. 718, pp. 244–251. Springer, Heidelberg (1993)
7. Fábrega, F.J.T., Herzog, J.C., Guttman, J.D.: Strand spaces: Proving security protocols correct. J. Comp. Secur. 7(2), 191–230 (1999)
8. Jonker, H.L., Pieters, W.: Receipt-freeness as a special case of anonymity in epistemic logic. In: Proceedings of the IAVoSS Workshop on Trustworthy Elections (2006)
9. Liu, Y.: Fairness analysis of e-commerce protocols based on strand spaces. International Journal of Grid and Utility Computing 4(2), 128–133 (2013)
10. Petrakos, N., Kotzanikolaou, P., Douligeris, C.: Using Strand Space Model to Verify the Privacy Properties of a Fair Anonymous Authentication Scheme. In: 16th Panhellenic Conference on Informatics, pp. 105–110 (2012)
11. Zhang, L., Luo, J.: Formal analysis of anonymity based on strand space model. In: First IEEE International Conference on Ubi-Media Computing, pp. 75–81 (2008)
12. Zuzana, R.: Electronic Voting Schemes. Master's thesis. Comenius University (2002)

Seamless Handoff of Ping-Pong Calls in Mobile Wimax Networks

Balu Sridevi[1], G.V. Shirley Julia[1], and S. Rajaram[2]

[1] Department of ECE, Velammal College of Engineering and Technology
[2] Department of ECE, Thiagarajar College of Engineering, India
aisveriya@yahoo.com, juliashirley@gmail.com,
rajaram_siva@tce.edu

Abstract. Wireless Networking is a promising technology that allows users to access a broad range of information and applications. The global boom in the number of users of the global internet has led to the development of fixed and mobile broadband technologies providing support for high speed streaming multimedia and unhampered quality of service. WiMAX technology given by IEEE 802.16e is one such technology that offers promising features in terms of high bandwidth, extended coverage area and low cost. WiMAX also suffers from certain issues like handoff delay and security threats. The ping-pong effect occurs due to the frequent movement of mobile units between the two Base Stations (BS) and thus experiences unwanted handoff delay at both BS. This paper proposes the algorithm for identification of ping-pong calls from normal re-entries, process them separately such that the handoff delay and resource wastage due to ping-pong calls are reduced. The proposed algorithm deals with caching of the key and uplink mapping parameters for the ping-pong users in order to reduce the overhead of the long network entry process. The network model was developed using Network Simulator and the algorithm was implemented in MATLAB GUIDE which gets connected to the database developed in MYSQL.

Keywords: WiMAX, Handover, Ping-pong, Authentication, IEEE 802.16e.

1 Introduction

Mobile communications services pierces into our society at an explosive growth rate. Achieving communication services at anytime, anywhere is considered as the evolution of communication technologies from wired to wireless with the miniaturization of devices. Currently WiMAX is developed in the same motive of enhancing it according to the needs of users. WiMAX is ahead of other technologies, since it would operate similar to Wi-Fi but at higher speeds, over larger distances and for a greater number of users. Current scenarios of wireless communication necessitate both security and speed with equal importance. Compromise on any one of this leads to the degradation of technology Seamless mobility of WiMAX is stressed due to handoff delays in Cross layer, Network layer, and MAC layer. Observations show that most of the users

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 428–439, 2014.

re-entering the base station frequently are ping-pong calls which cause system instability, call drop increasing and QoS degradation due to unnecessary processing of hand over at both base stations Since the ping-pong calls increase the occurrences of handover and thus overloads the network, it is necessary for network providers to reduce this undesirable effect. Ping-pong is defined as the case that the MSS, while moving to a Target BS and performing network re-entry procedures with the Target BS, tries to return to its Serving BS and resume the communication.

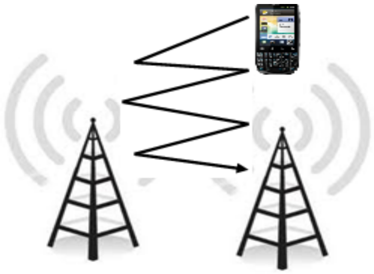


Fig. 1. Ping-pong Calls

2 Related Works

The handoff delay reduction has been an important area of research. One such research [10] comes forward with a solution to this issue, according to which a key caching mechanism is adapted to do away with the dispensable IEEE 802.1X authentication cost in a WiMAX handoff along with a study on how to assign time taken for key caching. To reduce the burden of database due to key caching and key exchange compression technique can be applied [9]. As the ping-pong calls suffer from more handoff delay and burden both the TBS and SBS for the same procedure, solving them is considered as a vital research issue. The work done in [5] and [11] discussed the method of increasing some thresholds in detecting and controlling the ping-pong handoffs. This results in decisions that increase the probability of late and failed handovers causing service interruption.

Research [6] demonstrated an effective way to classify ping-pongs in mobile broadband networks. It presents a method that can significantly reduce unwanted ping-pongs in the network. The method combines a sub cell movement detection method and ping-pong detection to decide when it is most effective to apply handover threshold tuning (pinning) without increasing the risk of late or failed handovers. Handoff prediction techniques [12] and [13] are used for reducing ping-pong occurrences. The shortcoming of existing ping-pong reduction solutions is that all ping-pongs are treated the same i.e., they are considered bad and should be eliminated. In reality ping-pongs are not equal and the solution to the problem could be optimized better by analyzing the ping-pong situation deeper.

Researches in this area have resulted in a mechanism in which the TBS, upon learning about the ping-pong effect, informs the previous SBS about the MS's reverting back to it [14].This helped the previous SBS to identify the return of the MS as an effect of ping-pong and not as an altogether new network entry. So, provided the SBS has retained the MS's previous connection information, call drop as well as ping-pong resumed quickly as the MS could get access to non- contentious ranging slots. However, this scheme will not work if the SBS has not retained the state information of the MS.

3 Existing Work

Ping-pong handover is a potentially undesirable phenomenon, in which the mobile station (MS) performs frequent transition between the same pair of cells back and forth within a short time period.

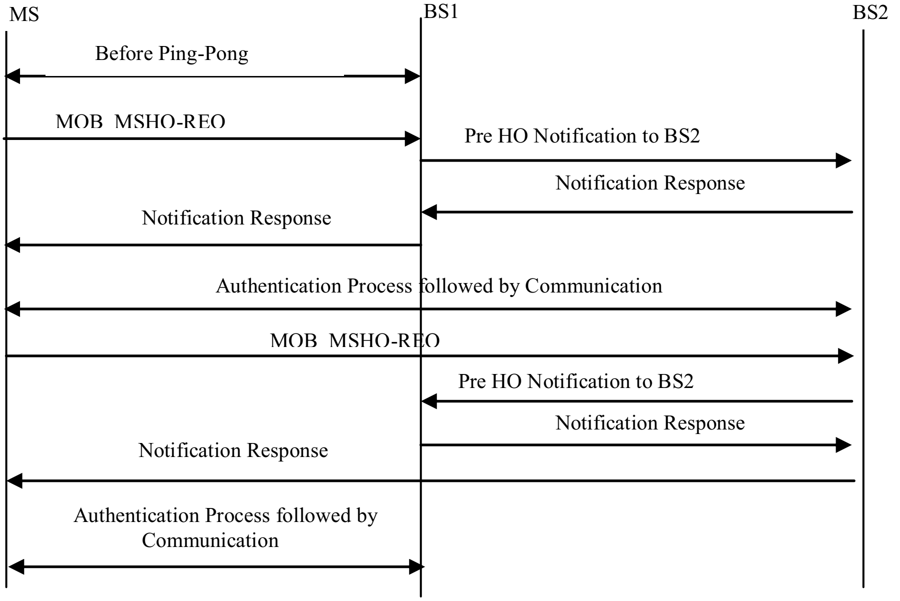

MS
BS1
BS2
Before Ping-Pong
MOB MSHO-REO
Pre HO Notification to BS2
Notification Response
Notification Response
Authentication Process followed by Communication
MOB MSHO-REO
Pre HO Notification to BS2
Notification Response
Notification Response
Authentication Process followed by Communication

Fig. 2. Existing Ping-pong Process

Reduction of undesired ping-pong handovers is an important task of mobile network management. The extra capacity required to serve a large number of ping-pong handovers comes with a non-negligible cost. The amount of ping-pong type handovers may account for approximately 40-60% of all handovers based on measurements in numerous networks. Another negative aspect of ping-pongs is their potentially adverse effect on mobile broadband services. When the MS switches between two BSs, transmission is delayed similar to the delay handover of any MS entering into BS. The existing process is shown in Fig. 2. in which every time when a Mobile Station (MS) changes from one Serving Base Station (SBS) to Target BS (TBS) the entire handover procedure consisting of Network Topology Acquisition Phase (NTAP) and the Actual Handoff Phase (AHOP) happens. Since there is no separate procedure for the processing of such ping-pongs, MS expects the handover to take place between two BS repeatedly. It is clearly known that the same handover process is duplicated many times overloading the resources of BS which also shares the functionality of BS to other MSs.

4 Proposed Work

The proposed work concentrates on how to process the ping-pongs calls resume faster rather than avoiding them. This work focused on identification of MS initiating ping-pong calls and processes them separately according to their arrival time. Thus the proposed work reduces the burden of Mobile WiMAX by fast resuming of ping-pong calls and saves the resources and time for other calls. This process is accomplished in four stages. The first stage is the registration of ping-pong user in the SBS. Second stage is the identification of ping-pong by MS itself followed by request for fast handover to finalize TBS through SBS. The third stage is the verification of ping-pong MS by SBS followed by forwarding of MS request to TBS or discarding. The final stage is the verification of ping-pong MS done by TBS followed by permitting ping-pong call or discarding.

4.1 Registration of Ping-Pong MS with BS

As explained earlier ping-pong user occurs due to frequent movements of MS between two BS. When the MS enters the BS, the handover procedure is done. But when the MS re-enters the BS, count which is initialized to 0 is incremented by 1 for each of its re-entry into the network as explained in Algorithm 1. Difference in time between its successive entries to the BS is computed and termed as Inter Arrival Time (IAT). Each BS contains both the list of neighboring BS to where Pre-Handoff Notification request Sent PHNS () and from where the Pre-Handoff Notification request Received PHNR () for each MS. If the call is ping-pong then the SBS which is forwarding the ping-pong request for MS will be the BS to which the handoff was already requested for the same MS. Even though it is proved to be ping-pong, fast handoff facility can be enjoyed from its third entry. In its second entry, the registration process is performed only after verifying whether IAT is less than T_{KCTP}, maximum Key Caching Time for ping-pong which will be greater than Resource Reservation Time T_{RRT}. This verification is done only for registration of ping-pong user. After registration MS are served based on their expiry time of cached details like T_{KCTP} and T_{RRT} as per its request. Then the details of MS will be inserted in the separate table for ping-pong calls. The details include the MAC address of the MS (MS_MAC), details of resource channels used in the previous entry of the same MS contained in Uplink Map Information Elements (ULMAPIE) and Traffic Encryption

Key (TEK) which is the final key used for encrypting data TEK. The uplink map provides the sub channel and slot allocation and other control information for the uplink sub frame. UL-MAP contains as many Information Elements (IE) as the number of data bursts. Each IE has a one-to-one correspondence to a user's data burst. Storing this information in the database for all re-entering MS will cause denial of service due to dumping of the database and also prone to various security attacks. Hence it is essential to fix timing for the storage of these details. Accordingly, TEK is stored for Key Caching Time T_{KCT} and ULMAPIE is stored for Resource Reservation Time T_{RRT} which is less than Key Caching Time for TEK of Ping-pong T_{KCTP}. After their expiry of the time, the stored details will be automatically deleted from the database. In the algorithm T(SBS_i) represents the time at which MS is released from SBS and TEKASN represents the TEK cached in ASN database.

Algorithm 1. Registration of Ping-Pong Users in BS

```
count=count +1
SIAT =Difference in time between current entry and previous entry of MS
//Initially count=0 and Incremented by 1 for each entry
If SBS_i == PHNS(MS)
   If count=2 and IAT ≤ T_KCTP seconds
        Insert MAC address, ULMAPIE and TEK of MS in a separate ping-pong table
        start session
        /*retain process*/
        Retain TEKASN(S) for T_KCTP Seconds
        Retain ULMAPIE(S) for T_RRT Seconds
     end
  end
```

4.2 Identification of Ping-Pong by MS

This stage helps the MS to identify whether the handoff is ping-pong and to initiate the MOB_MSHO request accordingly. If identified as ping-pong, MS analyze and categorize its request to TBS. After finalizing the TBS to which MS has to handoff, it checks whether it is same as the previous SBS who have attended it and then forwarded it previous handoff request. If it is same then it is proved as ping-pong call request otherwise decided as handoff request. To analyze the category or type of ping-pong it computes the difference in time between current time and the time at which it left the previous SBS or TBS. As explained in algorithm 2, if the difference is less than T_{KCT} then it can send ping-pong requests before key caching time (PINGPONG_KCT). When the difference is greater than T_{KCTP} and less than T_{RRT}, then it can send ping-pong requests before resource retain time (PINGPONG_RRT). If both the above conditions fail, it sends simply ping-pong request (PINGPONG).

Proposed work thus serves the ping-pong request even after its expiry which was not solved and addresses in any of the present researches. The above mentioned type of request should be sent to SBS and then forwarded to the finalized TBS. Depending on the type of request, MS can save the handoff time.

Algorithm 2.

```
//Handoff Initiation
//Negotiate BS capabilities and finalize the TBS
If TBSi==$SBS_{i-1}$
    If CurrentTime – $T(SBS_{i-1}) <T_{KCTP}$
        Send ‘PINGPONG_KCT’ Request to $SBS_i$
    else if   CurrentTime – $T(SBS_{i-1}) <T_{RRT}$
        Send ‘PINGPONG_RRT’ Request to $TBS_i$ through $SBS_i$
    else
        Send ‘PINGPONG’ Request to $TBS_i$   through $SBS_i$
    end
end
```

4.3 Verification and Forwarding of Ping-Pong Request from MS to TBS by SBS

Independent of the type of handoff whether it is MS initiated or BS initiated, the handoff request has to be processed only by SBS. On receiving a ping-pong request from MS, SBS verifies whether the request is valid by checking the existence of TBS in the list of BS who already forwarded the request of MS for handoff. If exists then forward the request to TBS as well as it acknowledge the MS. But it discards the request when the match not found as shown in algorithm 3.

ALGORITHM 3.

```
//Checks the list Pre-Handoff Notification Requests received from various BSs
if $TBS_i$ ==PHNR($MS_i$)
        Forward ‘PINGPONG_KCT’/‘PINGPONG_RRT’/‘PINGPONG’ to $TBS_i$
        Acknowledge $MS_i$
else
        Discard Request
end
```

Steps involving the authentication [15] of MS handoff to TBS need the role of SBS in forwarding the request from MS to TBS and responses from TBS to MS. The proposed work reduces the burden of SBS by skipping the authentication steps mentioned in [15] according to the type of ping-pong thus enhances the efficient utilization of BS by MSs.

4.4 Processing of Ping-Pong Request by TBS

After receiving the ping-pong request from MS through SBS, TBS verifies the validity of MS. If the MS is valid then the verification of its ping-pong request is done by checking whether the SBS is same as the BS to which handoff request was sent for the MS. Number of entries of MS to the BS is counted and it should be greater than two for processing the request. This is to ensure that the registration of ping-pong MS is done at its second arrival and the proposed methodology can be utilized in the further arrivals of MS. If both the mentioned conditions are satisfied, then the type of ping-pong request is considered.

Algorithm 4.

```
  //Check the validity of MS who sends PINGPONG Request   // Verification of MS by BS
if MS ∈U
   Verify MS_MAC, SAID, SigMS
     if verification succeed
          if SBS_i ==PHNS(MS) and Count>2
   .          if TEKASN(MS)≠ NULL   AND REQ=’PINGPONG_KCT’
                /*Handoff process till TEK Generation is bypassed by providing  TEK */
                TEK= TEKASN(MS)
                start session
                retaining process
              else if  ULMAP(MS)≠ NULL AND (REQ=’PINGPONG_RRT’
                   OR ’PINGPONG_KCT’ )
                /*Handoff process till ULMAPIE allocation is bypassed by   providing
ULMAPIE*/
                 ULMAPIE = ULMAPIE(S)
                     start session
                 retaining process
              else
                     /*Allocate a highest prioritized non-contention ranging  opportunity*/

             Fast_UL_Ranging
              retaining process
               end
         else
            Do not allow as Ping-pong user
            Check for re-entry process
         end
  else
      Invalid User
     end
```

The idea behind this technique as explained in Algorithm 4 is

1. To provide retained TEK to the ping-pong users who request for handoff within T_{KCT} which enables the MS to bypass up to 6 steps in handoff process [15].
2. To provide retained resources and channel details to the ping-pong users who request for handoff within T_{RRT} after T_{KCTP} which enables the MS to bypass up to 5 steps in handoff process [15].
3. To grant highest priority to the MS in acquiring demanded channels in spite of its arrival after its expiry.

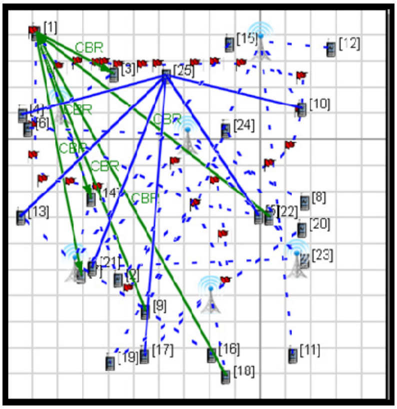
Above mentioned restrictions are made to maintain the database without overloading. Even though MS requests for PINGPONG_KCT, TBS cannot provide TEK if the request was not received within T_{KCT}. A similar process is done for PINGPONG_RRT. Any type of request comes from MS after T_{RRT} can be given higher priority in processing and allotment of ranging slots as desired by MS which is termed as Fast_UL_Ranging. After the processing of ping-pong requests, the details stored in the database are again stored and maintained. If the incoming MS is not proved to be ping-pong user, then it has to go undergo the re-entry procedure.

5 Results and Discussion

A real time analysis of MS re-entries is recorded from the mobile WiMAX network provided by the service provider AIRTEL in Madurai city. The dataset includes the arrival rate, identity and arrival time of mobile stations, Base station identities for handoff, etc. Handoff details for 6 Base stations are taken for the analysis. The average re-entries are estimated in the range of 10.4% to 31.3% based on data retrieved at random time. The real time analysis is used only as a proof for the possibility of MS re-entries and the research are continued with simulation works.

5.1 Qualnet Simulation Model

The scenario of this work is developed in Qualnet, algorithms were implemented in MATLAB 2010 and database used is MYSQL. The initially WiMAX environment is developed in Qualnet with 6 Base stations (BSs) each with coverage of 50 kms. Variable number of mobile stations (MSs) is placed randomly within the network as shown in Fig. 3. Each MS is made to move randomly and outputs were studied at every minute up to 10 mins within the network which obviously experiences handoffs and re-entries with BSs were noted. Performance analysis of ping-pong calls occurring in each BS is shown in Fig. 4 for all the four simulations. It shows that % of ping-pong calls are nearly constant in all simulations. Averages of ping-pong in each simulation are 15.48, 16.67, 17.25 and 16.24.



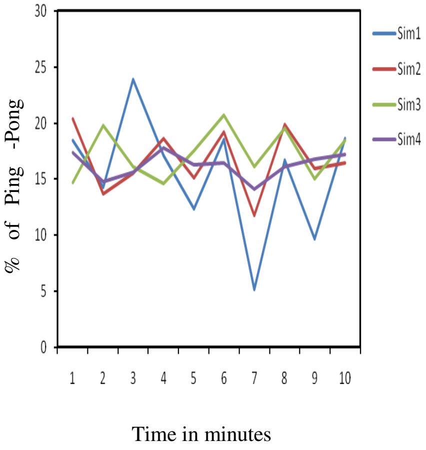
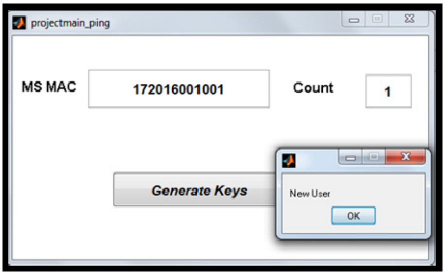


Fig. 3. Qualnet Simulation Model

Fig. 4. Average Ping-pong Calls

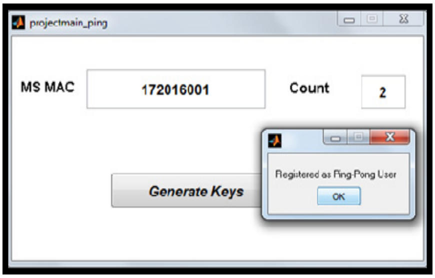
5.2 Proposed Ping-Pong Types and Its Processing

Implementation of three types of proposed ping-pong processes is demonstrated in this section. As explained in algorithm 1 first entry of MS into the BS is processed. The entry of MS into the BS is counted and shown in Fig. 5. When the verification of validity is satisfied, the keys are generated and AK is cached according to the type of key caching of MS allowed.


projectmain_ping

MS MAC 172016001001 Count 1

New User


projectmain_ping

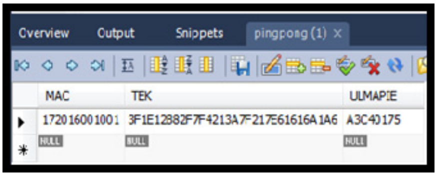
MS MAC 172016001 Count 2

Registered as Ping-Pong User

Fig. 5. MS entry into BS

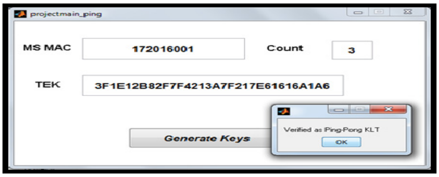
Fig. 6. Registration of MS as Ping-pong user

Validation of ping-pong is conducted to provide the fast handoff facility in its future entries according to algorithm 1. To attain this, the inter arrival time (IAT) between the first and second entry of MS is computed and if it is less than T_{KCTP}, then the registration of the MS is made as shown in Fig. 6. The details of the MS are stored in a separate database as a part of registration as shown in Fig. 7. TEK is stored for T_{KCT} and ULMAPIE is stored for T_{RRT}. As T_{RRT} is greater than T_{KCTP}, details of the resources of the last visit are maintained for more time compared to TEK. MS when enters as a ping-pong call beyond T_{RRT}, it cannot be provided with any details because of the automatic deletion of TEK and ULMAPIE soon after the expiry of T_{KCTP} and T_{RRT} respectively.



	MAC	TEK	ULMAPIE
▸	172016001001	3F1E12B82F7F4213A7F217E61616A1A6	A3C40175
*	NULL	NULL	NULL

Fig. 7. Caching of TEK and ULMAPIE



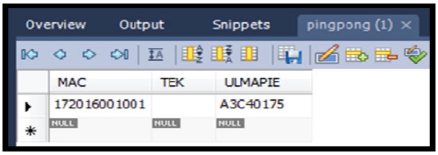
MS MAC 172016001 Count 3

TEK 3F1E12B82F7F4213A7F217E61616A1A6

Verified as Ping-Pong KLT

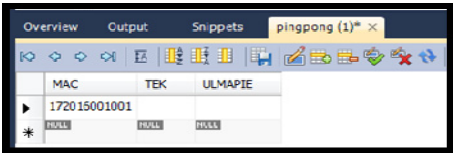
Fig. 8. PINGPONG_KCT Request

When the MS enters the BS for the third time or more, it is first verified as ping-pong call and then categorized according to the request sent by MS and the available details in the database as shown in Fig. 8. Even though BS receives 'PINGPONG-KCT' request from MS, TEK can be provided only when it is not delayed beyond $T_{KCT..}$



	MAC	TEK	ULMAPIE
▸	172016001001		A3C40175
*	NULL	NULL	NULL

Fig. 9. Deletion of TEK beyond T_{KCT}



	MAC	TEK	ULMAPIE
▸	172015001001		
*	NULL	NULL	NULL

Fig. 10. Expiry of TEK and ULMAPIE

Similarly 'PINGPONG-RRT' request from MS is verified and ULMAPIE is provided in order to render the resources retained in its last visit as shown in Figure 8 according to the algorithm 4. The point to be noted here is MS is provided with only ULMAPIE, even though it enters the BS before T_{KCTP}. Beyond the expiry of T_{RRT}, ULMAPIE is also deleted automatically from the database as shown in Fig.10.

5.3 Performance Analysis of Different Types of Ping-Pong Request

All types of users are treated as a new user in their first entry and old user in their second entry. The effect of the proposed technique is incorporated only from its third entry. From the figure it is evident that the MS is benefiting more when the request is sent before T_{KCTP}. Next comes the old user because of the caching of AK . Old user is followed by the MS who request before T_{RRT}. As discussed earlier when the MS sends the request beyond T_{RRT}, it cannot acquire TEKor AK or ULMAPIE. But it can be provided with high priority in allocating the ranging slots as requested faster. For the worst case, ping-pong user can avail this facility. In the real scenario of mobile WiMAX, users cannot be expected to arrive always in fixed timings. Assumed parameters of this simulation are TPP=20 sec's, T_{KCTP} =40 sec's, T_{KCT}=50 sec's and T_{RRT}=60 sec's. MS entered into the BS for 10 times each at random times and its consolidated handoff delay according to the mechanism shown in the algorithm .4 is computed. Performance analysis of handoff delay in 5such simulation is shown in Fig. 11.

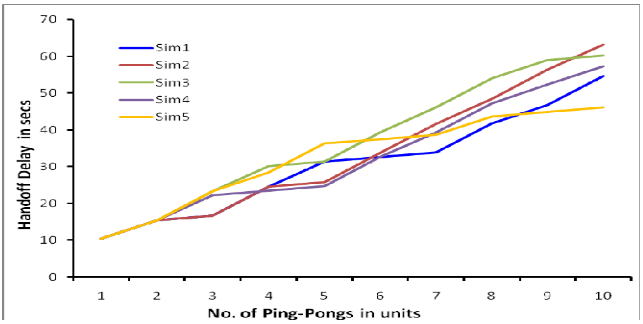


Fig. 11. Handoff delay of users with random arrival

Average Handoff delay reduction in each simulation and it is proved that the separate ping-pong processing reduces the handoff delay to the average of 46%.

6 Conclusion

The seamless service provided by the WiMAX network is often distorted by issues like handoff delay. This paper proposes an algorithm for addressing the handoff delay in processing Ping-pong calls. It is done by identification of MS initiating ping-pong calls and processes them separately according to their arrival time. Hence it helps in fast resuming of ping-pong calls and saves the resource and time for other calls. The Contributions of this paper are

- Identification of ping-pong users from normal re-entries'
- Reduction of handoff delay of separate processing of the ping-pong users identified.

Future enhancement of this paper is planned to analyze the performance of handoff delay reduction yielded by the proposed algorithms.

References

1. Andrews, J.G., Ghosh, A., Mohammed, R.: Fundamentals of WiMAX: Understanding Broadband Wireless Networking. Prentice Hall (2007)
2. Tang, S.-Y., Muller, P., Sharif, H.R.: Wimax security and quality of service. John Wiley & Sons Ltd. (2010)
3. Sridevi, B., Rajaram, S.: GUI based cost effective handoff management in the WiMAX Network entry process using key caching mechanism. In: SEISCON 2011. IEEE Explore (February 2012)
4. Hsu, S.F., Lin, Y.B.: A key caching mechanism for reducing WiMAX authentication cost in Handoff. IEEE Trans. Veh. Tech. 58(8) (October 2009)
5. Markopoulos, A., Pissaris, P., Kyriazakos, S., Sykas, E.D.: Efficient location-based hard handoff algorithms for cellular systems. In: Mitrou, N.M., Kontovasilis, K., Rouskas, G.N., Iliadis, I., Merakos, L. (eds.) NETWORKING 2004. LNCS, vol. 3042, pp. 476–489. Springer, Heidelberg (2004)

6. Feher, Z., Veres, A.: Ping-Pong Reduction Using Sub Cell Movement Detection. In: IEEE 75th Vehicular Technology Conference (VTC Spring), pp. 1–5 (2012)
7. Kang, H., et al.: Ping-pong Call Resuming Procedure during HO, IEEE 802.16Broadband Wireless Access Working Group Project, IEEE 802.16e/03-26r (2004)
8. Sridevi, B., Rajaram, S.: Compressed Key Exchange and Key Caching in PKMv2-EAP Mobile WiMAX Authentication. European Journal of Scientific and Research (March 2012)
9. Sridevi, B., Rajaram, S.: Dynamic Inter Arrival Time Based Seamless Handoff for Mobile WiMAX Ping-Pong Calls Bypassing PKMv2 EAP Authentication. International Journal of Computer Network and Information Security, 56–64 (June 2012)
10. Ray, S.K., Pawlikowski, K., Sirisena, H.: Handover in Mobile WiMAX Networks: The State of Art and Research Issues. IEEE Communications Surveys & Tutorials 12, 376–398 (2010)
11. Wang, S.S., Wu, C.-H.: Effective handoff method using mobile location information. In: Proceedings of the Fifty Third IEEE Conference on Vehicular Technology, pp. 2585–2589 (2001)
12. Becvar, Z., Mach, P., Simak, B.: Improvement of handover prediction in mobile WiMAX by using two thresholds. Journal of Computer Networks 55(16), 3759–3773 (2011)
13. Tseng, P.H., Feng, K.T.: 'A Predictive Movement Based Handover Algorithm for Broadband Wireless Networks'. In: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Las Vegas, USA, pp. 2834–2839 (2008)
14. Kang, H., Koo, C., Son, J., Lim, H.: Resource Remain type for Drop or Ping Pong Call Recovery, IEEE 802.16Broadband Wireless Access Working Group Project, IEEE 802.16e/03-26r1 (March 5, 2004a), http://www.ieee802.org/16/tge/contrib/C80216e-0426r1.pdf
15. Sridevi, B., Rajaram, S.: Automated secured handoff delay reduction by minimizing authentication cost in mobile WiMAX network entry process. International Journal of Innovation and Technology Management; (ISSN: 1741-5179 , Impact factor 0.727 - As indicated in Annexure I of refereed journal list given in Anna University Portal)

Static Malware Analysis Using Machine Learning Methods

Hiran V. Nath[1,2] and Babu M. Mehtre[1]

[1] Center for Information Assurance & Management,
Institute for Development and Research in Banking Technology, India
{hvnath,bmmehtre}@idrbt.ac.in
[2] School of Computer and Information Sciences (SCIS),
University of Hyderabad, India

Abstract. Malware analysis forms a critical component of cyber defense mechanism. In the last decade, lot of research has been done, using machine learning methods on both static as well as dynamic analysis. Since the aim and objective of malware developers have changed from just for fame to political espionage or financial gain, the malware is also getting evolved in its form, and infection methods. One of the latest form of malware is known as targeted malware, on which not much research has happened. Targeted malware, which is a superset of Advanced Persistent Threat (APT), is growing in its volume and complexity in recent years. Targeted Cyber attack (through targeted malware) plays an increasingly malicious role in disrupting the online social and financial systems. APTs are designed to steal corporate / national secrets and/or harm national/corporate interests. It is difficult to recognize targeted malware by antivirus, IDS, IPS and custom malware detection tools. Attackers leverage compelling social engineering techniques along with one or more zero day vulnerabilities for deploying APTs. Along with these, the recent introduction of Crypto locker and Ransom ware pose serious threats to organizations/nations as well as individuals. In this paper, we compare various machine-learning techniques used for analyzing malwares, focusing on static analysis.

Keywords: Malware, Static Analysis, Machine Learning, Advanced Persistent Threat, Cyber Defence.

1 Introduction

In the current scenario, the communication network forms a backbone of any industry. Any security breach, of these systems or networks is of major concern to the organization as well as for the society. Compromising of one or more of these leads to violation of confidentiality, integrity or availability. With the introduction of "IoT" Internet of Things, everything starting from household equipments like refrigerator to a host in mission critical areas (like a nuclear power plant or a pacemaker) are connected to internet. These devices could be

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 440–450, 2014.

controlled or monitored from anywhere around the world. Here security of all these devices plays a vital role.

Initially all research were focused on the analysis after the incident has occurred. Recently, the trend has been changed almost to a proactive approach, which is commonly known as Cyber Defense. Currently researchers are focusing on finding out methods to prevent the incident from occurring rather than analyzing the incident once it occurred. Here, they are using a predictive analysis method, by finding what would be the actions or methods that would be used by an attacker, to compromise the system.

Malware analysis is generally classified into two categories, Static and dynamic analysis. In static analysis, we would be classifying the file based on various features extracted from the executable. Disassembly is one of the methods, which is used for extracting various features from the executables. Here certain features are OPCODES, byte sequence, PE Header [30], etc. Shafiq at el. in [37] have described about a PE-Probe method through which we could detect previously unknown or zero-day malwares. Here the detection method works after classifying whether the malware is packed or not. They have claimed that it could be implemented for real-time analysis.

In dynamic analysis, we would be generally executing the malware in isolated environment for analysis. This could be a debugger or a virtualized or sandbox environment. This method is very much handy once the code is obfuscated. Here the analysis is based on the sequence of system calls initiated by the program. However, the drawback of this method is that, here not all the execution paths can be traced. Certain malwares are coded to behave in specific manner when they detect a debugger in the execution environment, or when these are executed in a virtualized environment. In case of obfuscated malware, each module would be decrypted only if that module is executed. So it would be very difficult to predict the actual behavior of the file.

In [36], Shabtai at el. have done a state of art survey on statically malware analysis. They have come to a decision that an individual classifier used on a single feature will not be useful. They have predicted that a weighted average of multiple classifiers on different features could give better results. Lot of research has been done on this field after 2009, and the methods used by attackers have improved a lot. Our aim is to analyze the current scenario, with the introduction of targeted and persistent attacks.

In section 2, we describe latest trends employed by malware developers, which is followed by various methods used in statical malware analysis categorized based on the features in section 3. In section 4, we discuss the impact of these methods on latest malware trends. Finally, in section 5 we conclude the paper by providing various types of malware attacks that is yet to be studied.

2 Recent Malware Trends

Recently, hackers are concentrating on compromising systems and networks for financial or political gains, rather than just for fame, which was their initial motivation. Cyber-attacks are initiated using highly sophisticated techniques, which is difficult to detect and defeat using current available methods. They are depending on multi-stage attacks for achieving their goals. Malicious software, known as malware, is one of the major tools used by these cyber criminals at one or more of these stages. These malwares can be of any form, ranging from integrated chips to Office or PDF documents attached to mail. So, preventive defense must be augmented for timely detection of attacks and initiation of responses. In this type of attack, usually the first stage will be using browser level exploits, like when you visit some malicious website, it executes some script or flash that launches some code, which is executed as a separate process. The second stage then downloads and runs the final payload onto the computer. This could be also considered as similar to that of k-ary codes, which is discussed later in this article. This accomplishes two objectives:

1. The attack is more likely to bypass some security checks or inspections by appearing less harmful.
2. The source of the attack is not easily identified by forensic analysis.

This model could be extended to any stage, in which the final or intermediate payload could be malware, which belongs to any of these families, like Zeus, Torpig, Gozi, Shylock or even a new one formed by zero-day exploits. These type of exploits are used to infect targets globally. The current infection rate is very high which proves the effectiveness of this multi-stage method.

Becker at el. [3], have proved that extremely stealthy hardware trojans can be embedded into integrated circuits, even below the gate level. Here they have inserted trojan by changing the dopant polarity of the existing transistors. Similarly, Lin at el. [22] have proposed about a hardware trojan which will induce physical side channel to convey secret information. In case of pdf documents even though its known that exploits could be written in Visual Basic Script and JavaScript's. Recently Filiol at el. [6] have shown that adobe portable document format by itself is a true programming language. This increases the risk associated with the PDF language based malware. Along with these, once the attacker uses code obfuscation, it becomes very hard for the researchers to analyze the programs. These methods are generally used for creating targeted attacks at multiple stages. In case of targeted attacks, the attacks are focused to specific targets like, an industry, or a country, or to accomplish a specific purpose like political espionage[21].

From 2009 onwards, there were series of targeted attacks, victimizing either an entire country or some particular industries. In most of these, malicious code has played a significant role. The published well-known targeted attacks are given bellow.

1. Aurora attack have targeted Google and was discovered in 2009[43].
2. Stuxnet or Flame or Duqu or Gauss targeted Iran's nuclear program, discovered in 2010 [26,4].
3. HTran or HUC (Honker Union of China) Packet Transmit Tool used for RSA breach, Found in 2011[27].
4. Madi or Mahdi had almost 800 victims in Iran & Israel[29].
5. Red October or IXESHE was targeted for Diplomatic, Governmental and Scientific Research Organizations[2,31].
6. MANDIANT targeted for Hong Kong[21].
7. ICEFOG targeted for Japan and South Korea[1].

The oldest known targeted attack was against Iran's nuclear program with well-known malware named Stuxnet. As per, McDonald at el. the C&C server used was registered on November 2005 for version 0.5 and the latest known version 1.x, was programmed to stop infection on June 2012. "ICEFOG" [1] is one of the most recent one, which was discovered in 26 September 2013. Here, they have used various known exploits like CVE-2012-1856, CVE-2012-0158, CVE-2013-0422 and CVE-2012-1723. In addition, they have used exploits in HLP (Windows Helper) and HWP exploits (Hangul Word Processor). This belongs to the category of targeted attacks since Hangul Word Processor is mainly used in South Korea, especially in the government sector.

These targeted attacks, form a superset of Advanced Persistent Threats (APTs)[40]. These APTs are multi-stage attacks, where multiple methods from social engineering to zero-day [28] exploit are used for accomplishing the task. Recently Dube at el. [11,9,10,12,13] have introduced a novel architecture which is known as Malware Target Recognition (MaTR). They claimed that their system is capable of capturing latest threats. Liu at el. [23] came out with an n-Victim approach to find out victims of APTs. Here they assumed that all the machines would be having similar network traffic due to C&C channel and, used N-gram method to analyze the network traffic.

Recently, there is a tremendous rise in the number of ransomware threats in internet. Many different variants of these threat are there, generally known as Cryptolocker and Ransomware. These threats hijack computer or its data and demand that a payment is made in order to unlock or decrypt them. The authors of these malicious threats have a strong financial motive for infecting as many computers as possible, and have put substantial resources in making these threats prevalent. New variants are being released frequently. Till date, there are no specific antivirus, capable of capturing Cryptolocker and Ransomware or their variants. If system gets infected and encrypted, they have to pay and wait for the mercy of the attacker/hacker. Symantec have come out with a report illustrating the percentage of infection, based on region. As per their document, greater than 10% of Indian community is infected with this malware.

Antivirus generally employee signature based methods for capturing malware. By performance analysis, it has been found that signature based system is very much inefficient compared to other static analysis methods. Current antivirus tools use sandbox or cloud environment for analyzing malware. In [14], Filiol has described k-ary malicious codes. Detection of k-ary codes is proved to be NP-Complete problem. Here k-ary means that instead of malware being delivered as a single file, it would be delivered as k files, which are mutually, disjoint set. It would be malicious only if all the k files are executed in a sequential or a parallel order. In [15], Filiol have came out with a method of combining k-ary codes with malicious cryptography techniques to amour code, which fails both static and dynamic reverse engineering techniques. A known example of this type of virus is 2-ary virus, which work in combination of W32.Qaz virus and W32.Funlove virus as referred in [16].

From long before itself, software or platform dependent malwares are there. Recently Vasiliadis in 2010 [41] have proposed about identifying and infecting the system based on the GPGPU card in that machine. More recently Denos at el. in [8] have proposed about a hardware depended malware, where it infects the system based on the onboard processor chips. Here the system works by identifying the family of processors' based on the Floating Point Arithmetic (FPA). These techniques are major stepping stones for hackers who are engaged in targeted attacks or APTs.

3 Methods for Static Analysis of Executables

Various data mining or machine learning methods are used for static analysis of malwares. The main bottleneck faced by researchers is the lack of enough number of training samples. The initial process for almost all of these methods is to obtain the disassembled code of the malicious program. Here the selection of feature and the machine-learning algorithm forms a unique method.

3.1 *n*-Gram

n-gram analysis is the method of analyzing byte sequences in a file. Here n stands for the number of bytes taken to form a single sample. n could take any values like 1,2,3,4... depending upon the designer. In [19,20,17,18], Kolter and maloof have came up with n-gram analysis. Here, they have fixed the value of n as 4 with a 1 byte sliding window after many trial and error. That is, instead of taking disjoint n-grams, they have taken over lapping n-gram patterns. From almost 1,651 malware samples, they able to create 255,904,403 distinct n-grams. Top 500 n-grams were selected as training set for analyzing the performance of various machine learning algorithms. Comparative study of naive Bayes, Support Vector Machine, Decision Tree (C 4.5) and its boosted version, by combining multiple classifiers were used. Here AdaBoost algorithm was used. The executables will be divided into ten disjoint sets, the first nine set were considered as training samples, and the final set was used as testing sample, for conducting ten-fold cross-validation. Here the best performance was given by boosted decision tree. They have obtained a true-positive rate of 0.98 for a desired false-positive rate of 0.05. This result was out-performed by MaTR architecture proposed by Dube at el.. For lower number of samples n-gram approach works fine, but as the number of sample increases the performance of the system decreases, but it stays better than the performance obtained by well-known antivirus.

In [44], Zhang at el. have also used n-gram as features based on the entropy (information gain). They have proposed a multi-classifiers system, which was built using probabilistic neural network. Here, they are using individual classifier to get the evidence, which is combined by Dempster-Shafer combination rules.

3.2 Byte Sequence

Information density or entropy is a method for measuring uncertainty in a series of numbers or bytes. Entropy measures the level of difficulty or the probability of independently predicting each numbers in a series. Here the basic assumption is that, if a file is encrypted, then its byte sequence would be very much scrambled, which in turn increases the information content of the file. This method is used mainly to identify the packed and encrypted malware. It was proposed by Lyda at el. in [25], where statically variation of malware executable is examined. Here a general assumption is taken like packing and encryption would be only used to conceal the code in a malicious executable. Lyda at el. have initially developed a bintropy, a binary file entropy analysis tool. Their experiments have shown that the entropy value would be high if the file is encrypted. Here they have analyzed 21,576 PE formatted tool. Here they have developed entropy metric, which could be generalized for any packed executables. This method is good for small size file whose size is less than 500KB.

3.3 OPCODE

An opcode stands for 'Operation Code'. An opcode is a single instruction that can be executed by the CPU. In machine language, it is a binary or hexadecimal value, which is loaded into the instruction register. In assembly language mnemonic, an opcode is a command such as MOV or ADD or JMP. Santos at el. in [33,32,34], have designed an machine learning based classifier which works based on frequency of appearance of opcode sequence. Here, they have used opcode sequence of length 2 and have discarded the parameters used by opcode. The assumption taken is that malicious operations take more than one machine code. Here they are using semi-supervised machine learning technique. It is very much useful when only small number of labeled data exit for each class. Roc-SVM method is used for partially labeled data. In their analysis, they have studies various cases like whether it is better to label malicious or being software, the optimal number of labeled instance and its impact on final accuracy and about how to reduce the labeling efforts. They have compared their results with simple Euclidean distance with malware labeled and with legitimate software labeled. They have also done 10-fold cross validation on 900 instances for each fold.

In [5], Bilar have used statistical data for detecting malicious executables. It was also based on the frequency distribution of opcode over malicious as well non-malicious samples. He has indicated that frequency of top five opcodes is same for malware as well as good-ware. They came out with 14 rare opcodes, like bt, fdvip, fild, fstcw, imul, int, nop, pushf, rdtsc, sbb, setb, setle, shld, std which could be considered as best feature for classification. These opcodes are more in malicious executables.

3.4 Portable Executable Header

Many researchers have used header information's extracted from portable executables as their feature set for classifiers. Majority of them works with assumption that the programs behavior could be predicted by analyzing PE Header details. This will store the information like the DLL's that the files is linked to. In 2001, Schultz at el. have came up with data mining approach for detecting new malicious executables [35]. The dataset he has analyzed was 3,265 malicious programs and 1,001 clean programs. Out of this data set, 38 malicious and 206 clean PE files were also analyzed. They have used list of DLLs used, list of DLL function calls, and number of different function calls per DLL as their feature set. They have used "Repeated Incremental Pruning to Produce Error Reduction" (RIPPER), which was proposed by Cohen in [7] as the classifier. The other classifiers they have used are Naive Bayes and Multi-Naive Bayes Classifier. This could be termed as the starting point for using header details for file classification.

PE-Probe. In [37] Shafiq at el., have came up with a system know as PE-Probe. Here the system works in two levels. The first level is a packer detector, which classifies between packed and non-packed executables and will be given as input to second level. Similar to other systems, for detecting packers, they uses features like number of standard, non-standard, executable section, read or write executable section, entries in import address table (IAT), along with the entropy of PE header, code section, data section, and entire PE File. This classification is done using multi-layer perceptron (MLP). Here, they are using PE-Miner from [39,38]. The PE-Miner was trained and tested on a large set of malicious executables files taken from VX-Heaven. In this, various preprocessing techniques like Redundant Feature Removal (RFR), Principal Component Analysis (PCA), Haar Wavelet Transform (HWT). The classification was done using five classifiers: instance based learner (IBk), decision tree (J48), Naive Bayes (NB), inductive rule learner (RIPPER), and support vector machines using sequential minimal optimization (SMO).Decision tree (J48) gave better detection accuracy compared to other classifiers. Here they have claimed to have better accuracy and performance than n-gram approach. The claim is that, *n-gram based techniques are more suitable for classification of loosely structured data; therefore, they fail to exploit format specific structural information of a PE file.*

Malware Target Recognition (MaTR). In this method, Dube at el. have claimed to have 98.5% efficiency. In [11,9,10,12,13], they have used bagged decision tree classifier. The classification was done based on the structural anomalies like, section names, section characteristics, entry point, imports, exports and alignment in PE Header. Here the prototype implementation was done using

TreeBagger in MATLAB. Here they have claimed to outdate Kolter and Maloof's n-gram approach. Their training set was 32-bit malwares obtained from VXHeaven until 2010. They have used 25,195 clean and 31,147 malicious samples. They have also claimed that they have tested the files, which belong to APT class, and the system is capable of detecting APT. However, they have not mentioned it clearly, whether their training set contains any APT or not. In addition, they have not mentioned what are the specific features with which they could capture targeted attacks.

Table 1. Comparison of various Static Malware Analysis Methods

Researcher	#Malware	#benign	Algorithm	Feature	True Positive	False Positive
Kolter & Maloof	1651	1971	Boosted Decision Tree	n-gram	0.98	0.05
Lyda & Hamrock	21567	0	Bintropy	Byte Sequence	—	0.005
Santos at el.	17000	—	Roc-SVM	Opcode-Sequence	0.96	0.05
Bilar	67	—	Frequency	Opcode	—	—
Shafiq at el.	—	—	Decision Tree	PE Header	0.996	0.003
Dube at el.	31147	25195	Boosted Decision Tree	PE Header	0996	—

4 Discussion

Initially in 1998, White have came out with various open problems in computer virus[42].Here he have considered computer virus as a similar one to that of biological virus. In 2006 Filiol at el. have came out with an updated version of various open problem in computer[16]. Here, they have indicated that, there is a lack of deep research in computer virus. Lot of the researchers think like research in this area is not of great use, since it is mostly a cycle of getting new sample, creating signature, updating the antivirus rather than having a general system, which is capable of capturing even new malware variants. Here, they have indicated various theoretical problems, which have yet to be addressed by research community.

The major bottleneck faced by researchers in using machine learning for malware analysis is the lack of training datasets. Each day new variant or encrypted version is released. Generally, it is not possible for the researchers to decrypt the encrypted code. Even if he were doing dynamic analysis for decrypting, only a certain subroutine would be decrypted and executed. Using that information, it is not possible to predict whether it is malware or not. So general scenario which happens, is to train the system to detect and check for obfuscation or encryption. If so, it is probably considered as a malware. So almost all the encrypted files generated by known packers, are also considered as a malware. This problem is equally applicable in Entropy Analysis for bytes sequence and MaTR Architecture. In entropy analysis, the bytes would be totally scrambled if the code is obfuscated. Obfuscation is used not only just for creating malicious code, but for hiding the logic or proprietary information in a program.

In case of code obfuscation, none of the opcode analysis methods will work. Since, it is only possible for the program to extract opcode from none obfuscated code blocks. That is, It is only possible for the system to extract opcodes from code block of the loader, which loads the encrypted file to memory. If we were training the system with this frequency of the loader, then all the files, which use this loader, would be considered as malicious code, which increase false positive. It leads to the same scenario given above.

In MaTR architecture, they have claimed that the system is capable of even detecting advanced threats. However, in their paper, they are not having enough proof to justify their claim. We have tried to obtain their code, so to test for the claimed efficiency, but we were not successful in getting the implementation details or the code. Since they have used decision tree approach, the efficiency depends on the selection of parameters, which is being considered in each level/node of the tree. Here, they are considering the header details, which mainly focus on Obfuscation. Therefore, there is a probability for the system to have more false positives even though their results shows promising.

Many researchers have claimed to use ensemble or boosted versions or multiple classifier one after another to detect malicious executables. However, none of them has done a comparative analysis on which sequence to select so that they would able to get a precise result. They have to also decide precisely on what are the features that should be considered on each level for each data-mining algorithm. The above results, shows that decision tree give good result for features like n-gram and PE header. It is necessary to have a good study of various features and algorithm combination to predict the sequence.

5 Conclusion

From the above study, we could conclude that none of the current research focus on a solution for multi-stage delivery of malware or *k-ary* codes or APTs. The trend of infection and exploiting has totally shifted from having a single malicious file to multi-stage or multiple files executed in parallel or sequential. Therefore, it would be better if we could have some machine learning system, which could correlate the possible activities of different files in parallel or sequential mode before classifying it as malicious or benign. None of above research deals with Crypto lockers and Ransom ware, which is one of the serious threats, now days. Therefore, there is a lot to be done in malware research to address these problems. Current solutions are not adequate for solving latest threats.

References

1. The ‘ICEFOG’ APT: A tale of cloak and three daggers. Kaspersky Lab Global Research And Analysis Team(GREAT) (2013)
2. Balduzzi, M., Ciangaglini, V., McArdle, R.: Targeted attacks detection with spunge. Trend Micro Research, EMEA (2013)

3. Becker, G.T., Regazzoni, F., Paar, C., Burleson, W.P.: Stealthy dopant-level hardware trojans (2013)
4. Bencsáth, B., Pék, G., Buttyán, L., Félegyházi, M.: The cousins of stuxnet: Duqu, flame, and gauss. Future Internet 4(4), 971–1003 (2012)
5. Bilar, D.: Opcodes as predictor for malware. International Journal of Electronic Security and Digital Forensics 1(2), 156–168 (2007)
6. Blonce, A., Filiol, E., Frayssignes, L.: Portable document format (pdf) security analysis and malware threats. Tech. rep., Virology and Cryptology Laboratory, French Army Signals Academy (2008)
7. Cohen, W.W.: Fast effective rule induction. ICML 95, 115–123 (1995)
8. Desnos, A., Erra, R., Filiol, E.: Processor-dependent malware... and codes. arXiv preprint arXiv:1011.1638 (2010)
9. Dube, T., Raines, R., Peterson, G., Bauer, K., Grimaila, M., Rogers, S.: Malware type recognition and cyber situational awareness. In: Second International Conference on Social Computing (SocialCom), pp. 938–943. IEEE (2010)
10. Dube, T., Raines, R., Peterson, G., Bauer, K., Grimaila, M., Rogers, S.: Malware target recognition via static heuristics. Computers & Security 31(1), 137–147 (2012)
11. Dube, T.E.: A Novel Malware Target Recognition Architecture for Enhanced Cyberspace Situation Awareness. Ph.D Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio (September 2011)
12. Dube, T.E., Raines, R.A., Grimaila, M.R., Bauer, K., Rogers, S.: Malware target recognition of unknown threats. IEEE Systems Journal 7(3) (September 2013)
13. Dube, T.E., Raines, R.A., Rogers, S.K.: Malware target recognition. US Patent 20, 120, 260, 342 (October 11, 2012)
14. Filiol, E.: Formalisation and implementation aspects of k-ary (malicious) codes. Journal in Computer Virology 3(2), 75–86 (2007)
15. Filiol, E.: Malicious cryptography techniques for unreversable (malicious or not) binaries. arXiv preprint arXiv:1009.4000 (2010)
16. Filiol, E., Helenius, M., Zanero, S.: Open problems in computer virology. Journal in Computer Virology 1(3-4), 55–66 (2006)
17. Kolter, J.Z., Maloof, M.A.: Learning to detect and classify malicious executables in the wild. The Journal of Machine Learning Research 7, 2721–2744 (2006)
18. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: An ensemble method for drifting concepts. The Journal of Machine Learning Research 8, 2755–2790 (2007)
19. Kolter, J.Z., Maloof, M.A.: Learning to detect malicious executables in the wild. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 470–478. ACM (2004)
20. Kolter, J.Z., Maloof, M.A.: Using additive expert ensembles to cope with concept drift. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 449–456. ACM (2005)
21. Li, F., Lai, A., Ddl, D.: Evidence of advanced persistent threat: A case study of malware for political espionage. In: 6th International Conference on Malicious and Unwanted Software (Malware), pp. 102–109. IEEE (2011)
22. Lin, L., Kasper, M., Güneysu, T., Paar, C., Burleson, W.: Trojan side-channels: Lightweight hardware trojans through side-channel engineering. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 382–395. Springer, Heidelberg (2009)
23. Liu, S.-T., Chen, Y.-M., Hung, H.-C.: N-victims: An approach to determine n-victims for apt investigations. In: Lee, D.H., Yung, M. (eds.) WISA 2012. LNCS, vol. 7690, pp. 226–240. Springer, Heidelberg (2012)
24. Lu, Y., Din, S., Zheng, C., Gao, B.: Using multi-feature and classifier ensembles to improve malware detection. Journal of CCIT 39(2), 57–72 (2010)

25. Lyda, R., Hamrock, J.: Using entropy analysis to find encrypted and packed malware. IEEE Security & Privacy 5(2), 40–45 (2007)
26. McDonald, G., Murchu, L.O., Doherty, S., Chien, E.: Stuxnet 0.5: The missing link. Symantec Security Response (online) 26 (2013)
27. Menn, J.: Key internet operator verisign hit by hackers. Reuters (February 2, 2012)
28. Muttik, I.: Zero-day malware. In: Virus Bulletin Conference (2010)
29. Prosecutors, Public: Messiah spyware infects middle east targets
30. Rafiq, N., Mao, Y.: Improving heuristics. In: Virus Bulletin Conference, pp. 9–12 (2008)
31. Raymond, D., Conti, G., Cross, T., Fanelli, R.: A control measure framework to limit collateral damage and propagation of cyber weapons. In: Fifth International Conference on Cyber Conflict (CyCon), pp. 1–16. IEEE (2013)
32. Santos, I., Brezo, F., Sanz, B., Laorden, C., Bringas, P.G.: Using opcode sequences in single-class learning to detect unknown malware. IET Information Security 5(4), 220–227 (2011)
33. Santos, I., Brezo, F., Ugarte-Pedrero, X., Bringas, P.G.: Opcode sequences as representation of executables for data-mining-based unknown malware detection. Information Sciences (2011)
34. Santos, I., Nieves, J., Bringas, P.G.: Semi-supervised learning for unknown malware detection. In: Abraham, A., Corchado, J.M., González, S.R., De Paz Santana, J.F. (eds.) International Symposium on DCAI. AISC, vol. 91, pp. 415–422. Springer, Heidelberg (2011)
35. Schultz, M.G., Eskin, E., Zadok, F., Stolfo, S.J.: Data mining methods for detection of new malicious executables. In: Proceedings of the 2001 IEEE Symposium on Security and Privacy, S&P 2001, pp. 38–49. IEEE (2001)
36. Shabtai, A., Moskovitch, R., Elovici, Y., Glezer, C.: Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey. Information Security Technical Report 14(1), 16–29 (2009)
37. Shafiq, M., Tabish, S., Farooq, M.: Pe-probe: leveraging packer detection and structural information to detect malicious portable executables. In: Proceedings of the Virus Bulletin Conference (VB), pp. 29–33 (2009)
38. Shafiq, M.Z., Tabish, S.M., Mirza, F., Farooq, M.: A framework for efficient mining of structural information to detect zero-day malicious portable executables. Tech. rep., TR-nexGINRC-2009-21 (January 2009), `http://www.nexginrc.org/papers/tr21-zubair.pdf`
39. Shafiq, M.Z., Tabish, S.M., Mirza, F., Farooq, M.: Pe-miner: mining structural information to detect malicious executables in realtime. In: Kirda, E., Jha, S., Balzarotti, D. (eds.) RAID 2009. LNCS, vol. 5758, pp. 121–141. Springer, Heidelberg (2009)
40. Sood, A., Enbody, R.: Targeted cyber attacks-a superset of advanced persistent threats. In: IEEE Computer and Reliability Societies, Michigan State University (2013)
41. Vasiliadis, G., Polychronakis, M., Ioannidis, S.: Gpu-assisted malware. In: 2010 5th International Conference on Malicious and Unwanted Software (MALWARE), pp. 1–6. IEEE (2010)
42. White, S.R.: Open problems in computer virus research. In: Virus Bulletin Conference (1998)
43. Zetter, K.: Google hack attack was ultra sophisticated, new details show. Wired Magazine 14 (2010)
44. Zhang, B., Yin, J., Hao, J., Zhang, D., Wang, S.: Malicious codes detection based on ensemble learning. In: Xiao, B., Yang, L.T., Ma, J., Muller-Schloer, C., Hua, Y. (eds.) ATC 2007. LNCS, vol. 4610, pp. 468–477. Springer, Heidelberg (2007)

Making an Application Provenance-Aware through UML – A General Scheme

P. Badharudheen[1], Anu Mary Chacko[2], and S.D. Madhu Kumar[2]

[1] Dept. of IT, MES College of Engg.,
Kuttippuram, Kerala
badharu@gmail.com
[2] Dept. of CSE, NIT Calicut,
Kerala, India
{anu.chacko,madhu}@nitc.ac.in

Abstract. An application is said to be provenance-aware when it monitors and captures the information regrading the activities of each and every process in that application. The provenance of a data item includes information about the processes and source data items that lead to its creation and current representation. This type of information or the metadata about the activities of each and every object in the application is very much important for the security purpose. The provenance information ensures the integrity of the data items and the objects involved in the application. Our approach enables an application to track the activities of each and every object involved in the application and captures the state changes of the objects into a permanent store. This information can later be queried by a Data Analyst whenever an attack by an intruder occurs into the application database. Majority of the provenance systems designed are domain specific. The methods available already for capturing the provenance are highly application specific. So there is a need to have a general methodology for capturing the provenance information automatically from the application while the application is under execution. In this paper, we present a general methodology for making an application provenance-aware by using the basic UML design diagrams of the application.

1 Introduction

In the context of application software, the word 'provenance' refers to the information about the basic objects that a particular object is depending on, the process of creation or modification of object states, etc. The concept of provenance and its utility has recently become increasingly prominent within the domain of information processing systems. Data Analysts or System Administrators who use data from a database or from any other sources are interested in information about the sources and the transformations that were applied to the data. They use this type of information as a source to track the changes made by an intruder and to re-configure the application to a safe state.

To determine the provenance of a data item or an object in a system, adequate documentation must be recorded regarding the processes with which it is involved,

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 451–460, 2014.

which is treated as the process documentation. The UML design diagrams of an application provide some information regarding the process documentation. This information can later be used for capturing the provenance information at the time of application execution. Here process documentation is comprised of multiple individual pieces of information, called *p-assertions* [7], which are captured during execution time of the application with the help of an A*pplication Tracer* and then stored and maintained in a repository of information called a P*rovenance Store*. This *Provenance Store* is initialised by some p-assertions which are captured from the UML design diagrams. To provide more security to the provenance information, the *Provenance Store* should be kept separate or isolated from the application repository so that an intruder cannot make attack to the provenance store easily.

The rest of the paper is outlined as follows. Section 2 presents the related work on provenance and these are categorized into four different types according to the way in which they capture the provenance information. In Section 3, we have described our proposed method for making an application provenance-aware. Section 4 explains the working of our *XMI-Parser* which is used to capture the initial process documentation details from the design diagram of an application. Section 5 presents the *Application Tracer*. In section 6, we have described the importance of provenance in security. Section 7 concludes the paper and outlines the future extensions of the ongoing work.

2 Related Work

There are two basic views of provenance. The first one describes the provenance of a data item as the processes that lead to its creation and the second focuses on the source data from which the data item was derived. Here we present some of the related works on provenance from the scientific and business domains.

The literature on provenance can be broadly classified into four categories: *fine granularity provenance systems*, *domain specific provenance systems*, *middleware provenance systems*, and *provenance in database* systems [8].

2.1 Fine Granularity Provenance System

The granularity of documentation means how detailed it captures the provenance information about a process is. For instance, if a system records all the instructions in a program, whereas another system records only the name of the program being executed, then we can say that the first system records documentation at a finer granularity. With finer granularity documentation, the corresponding representation of provenance will be more detailed. One example is the Transparent Result Caching (TREC) prototype as explained in paper [1]. TREC uses the Solaris UNIX proc system to intercept various UNIX system calls in order to build a dependency map.

A more comprehensive system which captures provenance is *audit facilities* implemented in the S language. S is an interactive system used for statistical analysis. Here it captures the result of users commands automatically in an audit file. The results include the creation or the modification of data objects as well as the commands executed.

2.2 Domain Specific Provenance Systems

Much of the research in the area of provenance comes in the context of domain specific applications. One of the major domains is the area of bioinformatics. The myGrid project as explained in the paper [2] has implemented a system for recording the documentation of process in the context of in-silico experiments.

GIS is another domain where provenance information can be recorded. In the paper [3], Lanter describes a system for recording process documentation and retrieving the provenance of map products in a GIS.

2.3 Middleware Provenance Systems

One of the major middleware provenance systems found in the literature is the Chimera Virtual Data System [4], which combines a Virtual Data Catalogue, for representing data derivation procedures, with a Virtual Data Language interpreter that translates user requests into data definition and query operations on the database. Using the Virtual Data Language, a user can query the catalogue to retrieve the Directed Acyclic Graph (DAG) of transformations which led to the result.

2.4 Provenance in Database Systems

In paper [5], Peter Buneman et al. define data provenance in the context of database systems as the description of the origins of data and the process by which it arrived at the database. "Why-provenance" refers to why a piece of data is in the database, i.e. what data sets or tuples contributed to a particular data item, whereas, "where-provenance" refers the location of the data element in the source data. Based on this terminology, a formal model of provenance is developed for both relational and XML databases.

Laura Chiticariu et al. define another model named DBNotes [6], a Post-It note system for relational databases where every piece of data may be associated with notes or annotations. These annotations are transparently propagated along with the actual data as it is being transformed. For instance, the annotations associated with a piece of data 'd' in the result of a transformation consist of the annotations associated with each piece of data in the source where 'd' is created from.

3 Making an Application Provenance-Aware Using UML

UML can be described as a general purpose modelling language to visualise, specify, construct and document software systems. We all know that every application development process starts by drawing its design diagrams first. So by critically evaluating the design diagrams of the applications, we will be able to identify the modifications needed for application to make it provenance-aware.

In order to make an application provenance-aware, we have to start from the design phase of the application. This phase itself gives some information needed for application to make it provenance-aware. The proposed method for making an application provenance-aware uses the concept of object oriented programming where objects are the main actors of any activity in an application. A method by Simon Miles et al. in paper [7] describes a similar but different approach for making an application provenance-aware using actors and their interaction in a system.

The proposed method for making an application provenance-aware uses the following initial assumptions.

- Applications are composed of objects and their interactions.
- Objects have attributes and their own operations or methods.
- Each object communicates with other objects by exchanging messages or through interactions. They are the stimuli for any operations in the application.
- The design diagrams of the application should be developed by using any standard tools like ArgoUML, Umbrello, etc.

Following are the three phases used to capture the provenance from the basic UML design diagrams [8].

3.1 Design Analysis

In the design analysis phase, our system analyses the UML design diagrams of an application and identifies the information needed for application to make it provenance-aware. By critically analysing the UML class diagrams of an application, we can identify the different attributes and operations or method calls of each class in the application. These method calls or interactions are the stimuli for any operation in the application.

By analysing the initial object diagram, identify the different objects and the initial states of each object. These object states will be changed later when an interaction occurs among the objects. If we are making the design diagrams of the application by using any standard design tools, we can easily parse the output design diagram with the help of an *XMI-Parser*. Our *XMI-Parser* is responsible for recording the initial process documentation details from the design diagrams. This initial process documentation details or the initial state of each and every object in the application are treated as the initial provenance information.

3.2 System Initialisation

In the system initialisation phase, the proposed system uses three permanent stores named *Opstore*, *Objstore* and *Provenance Store*. It initialises the *Opstore* by keeping all the operations and it's details identified in the first phase. Also it initialises the *Objstore* by keeping all the objects and its details identified in the first phase. The Provenance Store is initialized with the attribute values or the states of each objects identified before. The *XMI-Parser* is responsible for defining and initiating the structure of the above three permanent stores. These permanent stores should be kept separate from the application repository so that an intruder cannot make attack to the provenance store easily. Isolating the permanent stores from the application storage area provides more security to the provenance information.

3.3 Provenance Collection

In this phase, the proposed system records the provenance information into the Provenance Store. Here our *Application Tracer* can be used for tracking the interactions among the objects in the application. This *Application Tracer* should be developed by the application developer as part of the application development. The inputs for the *Application Tracer* are taken from the *Opstore*. For each entry in the *Opstore*, it should record the attribute value changes of the objects and the details of interaction between the objects into the Provenance Store.

Also, if a new object is created, it should update the *Objstore* and the Provenance Store with the new object details. Figure 1 below shows the block diagram of the proposed method.

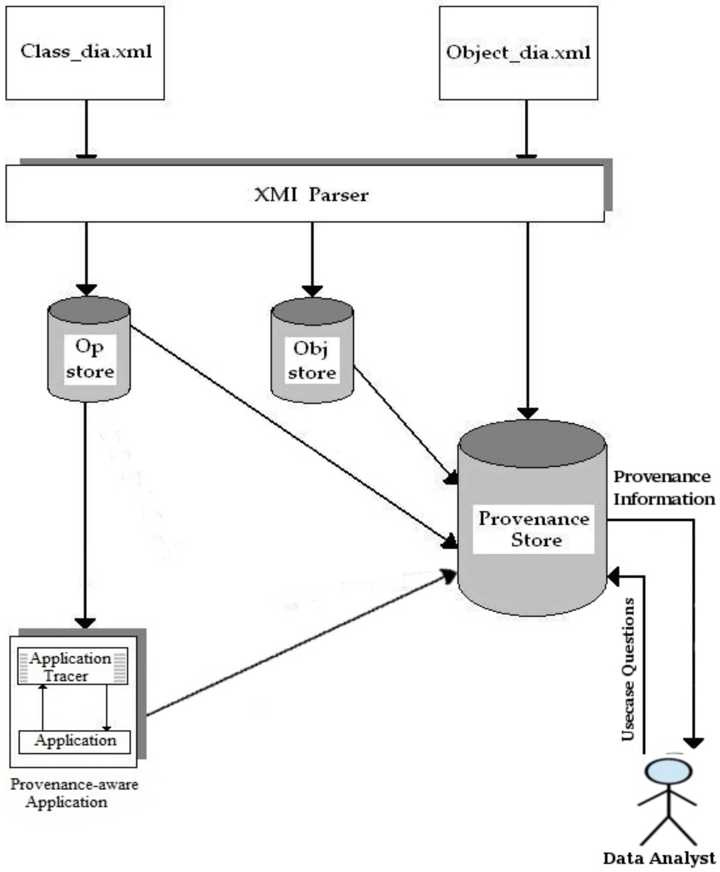


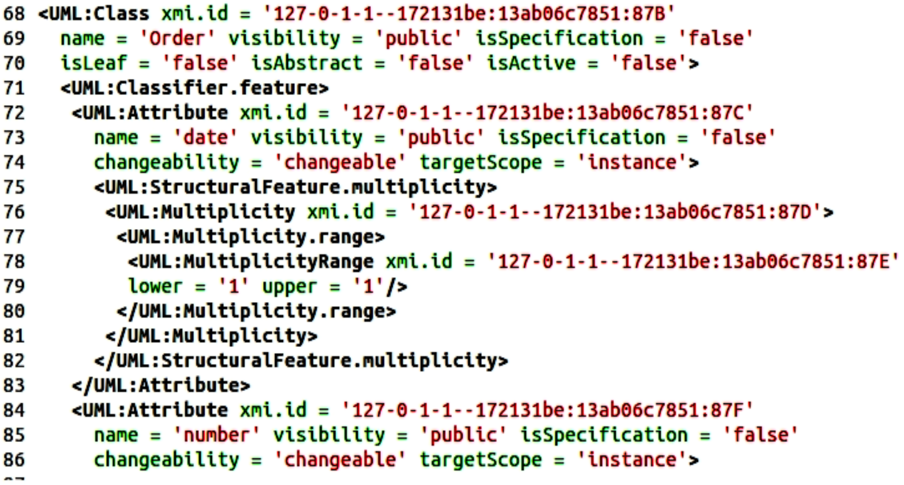
Fig. 1. Block diagram for the proposed method

4 The XMI-Parser

The XML Metadata Interchange (XMI) [9] is an Object Management Group (OMG) standard for exchanging metadata information via Extensible Markup Language (XML). The most common use of XMI is as an interchange format for UML models. Several UML tools like ArgoUML, Umbrello, etc. provide support for XMI files by importing or exporting their UML models in the XMI format.

Our *XMI-Parser* has two main functions. The first function is to define the internal structure of the three permanent stores used in the design. The second function is to parse the class diagram and the object diagram which we have given as inputs. The *XMI-Parser* then collects the initial process documentation details such as class names, initial objects with attribute states, different operations in the application, etc. by parsing the two design diagrams. It then records this process documentation details into the three permanent stores just defined before.

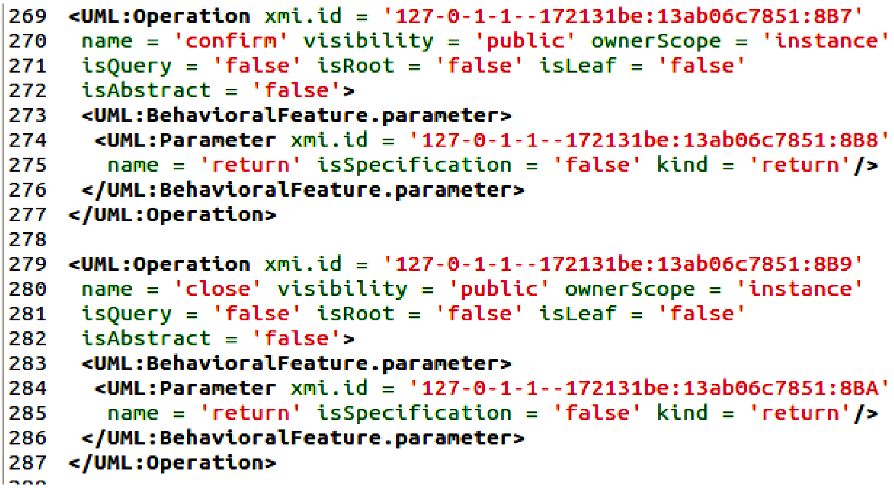
The *XMI-Parser* checks for certain predefined tags that are used to define the structure of the design diagram of an application. Here each component in the design diagram has a tag with some id which can be used to identify uniquely from other components in the diagram. These tags also have some attributes with values that are used to define the property of the components. Figure 2 below shows a sample code snippet from an xmi file of a class diagram.



```
<UML:Class xmi.id = '127-0-1-1--172131be:13ab06c7851:87B'
  name = 'Order' visibility = 'public' isSpecification = 'false'
  isLeaf = 'false' isAbstract = 'false' isActive = 'false'>
  <UML:Classifier.feature>
   <UML:Attribute xmi.id = '127-0-1-1--172131be:13ab06c7851:87C'
     name = 'date' visibility = 'public' isSpecification = 'false'
     changeability = 'changeable' targetScope = 'instance'>
     <UML:StructuralFeature.multiplicity>
      <UML:Multiplicity xmi.id = '127-0-1-1--172131be:13ab06c7851:87D'>
       <UML:Multiplicity.range>
        <UML:MultiplicityRange xmi.id = '127-0-1-1--172131be:13ab06c7851:87E'
        lower = '1' upper = '1'/>
       </UML:Multiplicity.range>
      </UML:Multiplicity>
     </UML:StructuralFeature.multiplicity>
   </UML:Attribute>
   <UML:Attribute xmi.id = '127-0-1-1--172131be:13ab06c7851:87F'
     name = 'number' visibility = 'public' isSpecification = 'false'
     changeability = 'changeable' targetScope = 'instance'>
```

Fig. 2. Sample code snippet from an XMI file of a class diagram

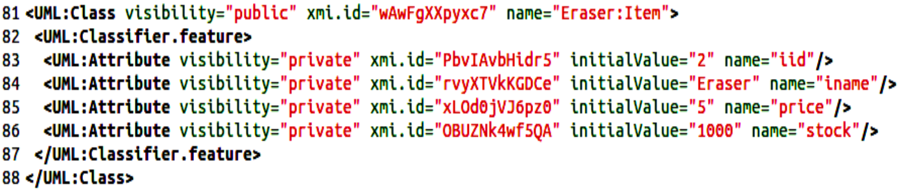
The xmi tag <*UML: Class*> is used to indicate the start of a class. One of the important attributes of this tag is the 'name' which indicates the class name. Under the <*UML: Class*> tag, it define the different attributes of this class by using the tag <*UML: Attribute*>. This tag also has some attributes like 'xmi.id', ‘name’, 'visibility', etc. given as arguments. The function of our *XMI-Parser* is to parse this xmi file and retrieve the values of attributes of the different tags. Also, from this class xmi file, we can find another important tag <*UML: Operation*> with an xmi id as same as that of the class in which this operation is defined. This tag has some important attributes like 'name' to indicate the operation or function name, 'visibility', etc. Figure 3 shows the details of this tag.



```
<UML:Operation xmi.id = '127-0-1-1--172131be:13ab06c7851:8B7'
 name = 'confirm' visibility = 'public' ownerScope = 'instance'
 isQuery = 'false' isRoot = 'false' isLeaf = 'false'
 isAbstract = 'false'>
 <UML:BehavioralFeature.parameter>
  <UML:Parameter xmi.id = '127-0-1-1--172131be:13ab06c7851:8B8'
   name = 'return' isSpecification = 'false' kind = 'return'/>
 </UML:BehavioralFeature.parameter>
</UML:Operation>

<UML:Operation xmi.id = '127-0-1-1--172131be:13ab06c7851:8B9'
 name = 'close' visibility = 'public' ownerScope = 'instance'
 isQuery = 'false' isRoot = 'false' isLeaf = 'false'
 isAbstract = 'false'>
 <UML:BehavioralFeature.parameter>
  <UML:Parameter xmi.id = '127-0-1-1--172131be:13ab06c7851:8BA'
   name = 'return' isSpecification = 'false' kind = 'return'/>
 </UML:BehavioralFeature.parameter>
</UML:Operation>
```

Fig. 3. Sample code snippet for the <*UML: Operation*> tag

In the same manner, The *XMI-Parser* parses an xmi file for the object diagram and collects the state values of different attributes of different objects into permanent store. This permanent store will be populated later at the time of application execution. Figure 4 below shows a sample code snippet of an xmi file of an object diagram.

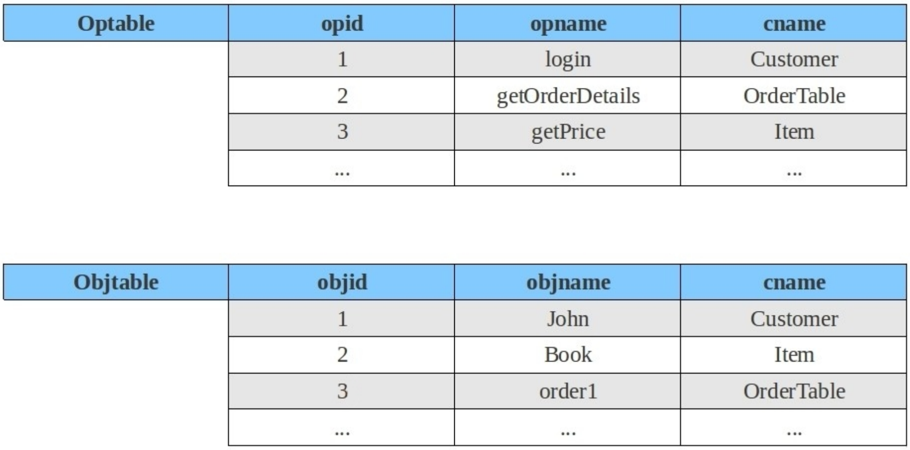


```
<UML:Class visibility="public" xmi.id="wAwFgXXpyxc7" name="Eraser:Item">
 <UML:Classifier.feature>
  <UML:Attribute visibility="private" xmi.id="PbvIAvbHidr5" initialValue="2" name="iid"/>
  <UML:Attribute visibility="private" xmi.id="rvyXTVkKGDCe" initialValue="Eraser" name="iname"/>
  <UML:Attribute visibility="private" xmi.id="xLOd0jVJ6pz0" initialValue="5" name="price"/>
  <UML:Attribute visibility="private" xmi.id="OBUZNk4wf5QA" initialValue="1000" name="stock"/>
 </UML:Classifier.feature>
</UML:Class>
```

Fig. 4. Sample code snippet from an XMI file of an object diagram

In figure 4, it uses a tag <*UML:Attribute*> with some arguments and their values to define the attribute of an object. So our *XMI-Parser* check for these tags in the xmi file and records the different attributes of an object with their current state value into the Provenance Store. This state values will be changed later at the time of application execution and are recorded into the Provenance Store with the help of an *Application Tracer*.

In the current design of our proposed method, we have included only the class diagram and the object diagram of an application. Our *XMI-Parser* has produced eminent result of initial provenance information by parsing these two diagrams of an application. As part of our initial experiment, we developed an online order management application and made it provenance-aware using our proposed method. Figure 5 below shows the internal structure of the object table and the operation table used in our sample order management application.

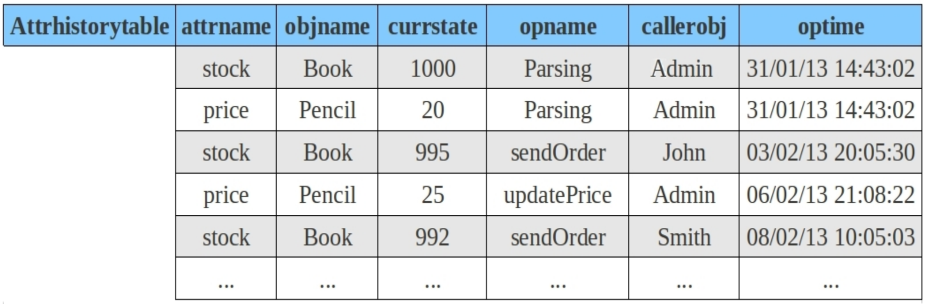


Optable	opid	opname	cname
	1	login	Customer
	2	getOrderDetails	OrderTable
	3	getPrice	Item
	...	...	...

Objtable	objid	objname	cname
	1	John	Customer
	2	Book	Item
	3	order1	OrderTable
	...	...	...

Fig. 5. The internal structure of the operation table and the object table

The first table shows the different class names and the operations defined in each class. The second table shows the class names and their initial objects. These two tables are populated at the time of parsing the design diagrams. We have defined another table '*Attrhistorytable*', for capturing the history log of the various operations performed in the application. It keeps the state changes of each attribute of each object in the application. The attribute states are normally changed when the application performs an operation defined in the *Optable*. So it keeps the corresponding operation name in the same table. These information can be treated as the provenance information. Figure 6 below shows the internal structure of the '*Attrhistorytable'*.



Attrhistorytable	attrname	objname	currstate	opname	callerobj	optime
	stock	Book	1000	Parsing	Admin	31/01/13 14:43:02
	price	Pencil	20	Parsing	Admin	31/01/13 14:43:02
	stock	Book	995	sendOrder	John	03/02/13 20:05:30
	price	Pencil	25	updatePrice	Admin	06/02/13 21:08:22
	stock	Book	992	sendOrder	Smith	08/02/13 10:05:03
	...	...	...	...	...	...

Fig. 6. The internal structure of the *Attrhistorytable* table

The Application tracer is responsible for updating the *Attrhistorytable* with the details of operation at the time of application execution. Our *XMI-Parser* initializes the above table with initial states of each object in the application.

5 The Application Tracer

The *Application Tracer* is responsible for capturing the provenance at the time of application execution. It provides a set of methods for the application developer so that the developer can directly include these methods in their application for capturing the provenance at the run time of the application. These *Application Tracer* methods are called from all the operations or function calls in the application which is already recorded in the *Opstore*.

It is the responsibility of the application developer to develop an *Application Tracer* as part of the application development. Here the *Application Tracer* traces the attribute state changes of each object in the application and records the corresponding changes in the permanent store as shown in Figure. 1.

6 Provenance for Security

Both provenance and security are closely related. Consider a hypothetical case of electronic voting system which collects the provenance information automatically for each and every activities in the voting system. Here the election officials and concerned citizens could simply review the provenance record to identify and explain any problems resulting from elections. More importantly, these records can be utilized to prove the activities of malfunction and potentially to recover from failed processes. Here, the advantages of a provenance record clearly cut across security and performance goals.

Another importance of provenance in security is in the implementation of access control mechanism to a particular data item. Here access for particular data item for a particular user can be granted based on the activity history. It can be a '*read*' access or a '*write*' access. So these history log details is very much important in making decisions about the access permissions. So provenance plays an important role in the protection of confidentiality.

The goals of a data item security include confidentiality, integrity, availability, and assurance. The provenance information of a data item will not ensure all these goals of security. But, these type of information can be used as a source to recover the original data when a security attack occurs. So, it is better to provide some high level of security to such type of data. If an intruder or a third party tries to attack into the provenance-aware application, our *Application Tracer* automatically captures the details of the corresponding operation in to the Provenance Store.

7 Conclusion and Future Work

In this paper, we have presented our innovative general scheme for making an application provenance-aware by using the basic UML design diagrams. The provenance information captured from any application are very much important to find out the root causes of security attack in the application. Also these information are essential in recovering the original data. So both provenance and security are closely related. In most contexts, when security matters, some data provenance information, if made available are found to be extremely useful.

We have described the scheme in three distinct phases *(i) design analysis phase*, where the design diagrams are analyzed for capturing the initial process documentation details *(ii) system initialisation phase*, where we use three permanent stores for initialising the process of capturing the provenance (*iii) provenance collection phase*, in which the provenance information are captured at run time of the application with the help of an *Application Tracer*.

Currently, in our design, we have included only the class diagrams and the object diagrams of an application. The provenance information obtained from the initial experiment using our proposed method was eminent. As an extension of this work, we plan to include other UML diagrams like sequence diagrams in the design for making our proposed method more efficient in capturing the provenance information.

References

[1] Vahdat, A., Anderson, T.: Transparent result caching. In: Proceedings of the Annual Conference on USENIX Annual Technical Conference, ATEC 1998, p. 3. USENIX Association, Berkeley (1998)

[2] Zhao, J., Goble, C., Greenwood, M., Wroe, C., Stevens, R.: Annotating, linking, and browsing provenance logs for e-science. In: Proc. of the Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data, pp. 158–176 (2003)

[3] Lanter, D.P.: Lineage in GIS: The Problem and a Solution. Technical paper. National Center for Geographic Information and Analysis (1990)

[4] Foster, I., Vockler, J., Wilde, M., Zhao, Y.: Chimera: a virtual data system for representing, querying, and automating data derivation. In: Proceedings of the 14th International Conference on Scientific and Statistical Database Management, pp. 37–46 (2002)

[5] Buneman, P., Khanna, S., Tan, W.-C.: Why and Where: A Characterization of Data Provenance. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, p. 316. Springer, Heidelberg (2000)

[6] Chiticariu, L., Tan, W.-C., Vijayvargiya, G.: DBNotes: a post-it system for relational databases based on provenance. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, SIGMOD 2005, pp. 942–944. ACM, New York (2005)

[7] Miles, S., Groth, P., Munroe, S., Moreau, L.: Prime: A Methodology for Developing Provenance-aware Applications. ACM Trans. Softw. Eng. Methodol. 20(3), 8:1–8:42 (2011)

[8] Badharudheen, P., Chacko, A.M., Madhukumar, S.D.: A Scheme for Generating Provenance-aware Applications Through UML. International Journal of Management and Information Technology 6(3) (2013)

[9] Lanceloti, L.A., Maldonado, J.C., Gimenes, I.M.S., Oliveira Jr., E.A.: Smartyparser: a xmi parser for uml-based software product line variability models. In: Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems, VaMoS 2013, pp. 10:1–10:5. ACM, New York (2013)

Enhancing E-Payment Security through Biometric Based Personal Authentication Using Steganography Scheme – B-PASS

Balasubramanian Chelliah[1] and S. Geetha[2]

[1] Department of Computer Science and Engineering,
P.S.R. Rengasamy College of Engineering, Sivakasi. Tamil Nadu, India
[2] Department of Information Technology,
Thiagarajar College of Engineering,
Madurai – 625 015, Tamil Nadu, India
geethabaalan@gmail.com

Abstract. Biometrics (or biometric authentication) which is more secure than conventional password based scheme, consists of methods for uniquely recognizing humans based upon one or more intrinsic physical or behavioral traits. Biometrics has become increasingly more valuable as a tool for verifying identities in a new and deeply interconnected national security environment. It plays a major role in almost all e-transactions. These e-transactions require a commitment to secure transactional details, including credit card information of the customers from various attacks such as replay attack, circumvention, repudiation and covert acquisition. To overcome the above mentioned attacks, a system that enhances E-payment security through Biometric PASS (Personal Authentication using Steganography Scheme) has been proposed. In this system a B-PASS card is generated by collecting the user's fingerprint and pin number during registration, which is later checked during verification phase. Transaction is possible only if all the three components (fingerprint, pin number and B-PASS card) are available and verified to be genuine. This system gives the internet users the confidence to perform e-transactions without the need to worry about the hackers or online shopping frauds.

Keywords: E-Payment Security, E-Commerce, Steganography, Biometric authentication, B-PASS.

1 Introduction

In today's Internet world, the rate of online financial transactions is increasing day by day. In India, the proliferation of E-transactions is predominant in all major financial sectors like railway E-reservation, online telephone bill payment, online EB bill payment, online tax payment, online LIC payment, E-shopping, E-recharge (any mobile service), internet banking, housing loan payments etc.

Gartner, Inc. [1] estimates that nearly 80 percent of business-to-business payments will be made electronically in 2014. With that number expected to grow to over 90

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 461–472, 2014.

percent by the year 2015, this causes a high pressure on managing directors, CEOs and CIO’s to create and implement e-business solutions currently. But as we operate today within a global networked economy that is dependent on the reliability and security of the information chain, this yields a complex challenge that requires a "culture of security" across industry sectors and the public infrastructure around the globe [2].

In all these transactions mentioned above which are made online over Internet, security is a major issue. The money must be received by the authentic services and correct amount of money has to be debited from the customers account. The money transactions are to be robust against many attacks like man-in-the-middle attack, phishing attack etc., which are very popular in the Internet. Biometric based authentication is a good solution which solves many security problems. These Biometric data like fingerprints, iris, palm print, face, voice, handwritten signature, etc. (as already proven) are unique for each individual and are considered to be more secure [3]. But still these Biometric data are suspicious towards the attacks like replay attack, circumvention, repudiation and covert acquisition.

To overcome the above mentioned attacks, a system for enhancing E-payment security using Fingerprint based Biometric PASS (Personal Authentication using Steganography Scheme) has been proposed in this paper.

2 E-Transaction Security Requirements and Security Threats

Security might differ from others in various points of views. However, there are some general security objectives to be considered by both merchants and customers during transaction. There are other security threats that affect the transaction process.

2.1 E-Transaction Security Threats

Traditionally, in data security dimension, four objectives are identified. They are confidentiality, integrity, availability and accountability [4]. Any E-Transaction system faces the following threats from the perspective of these four security objectives:

- **Threats to confidentiality:** On the customer's side the confidentiality can be compromised if cookies are collected at a central site, and a profile of the customer's browsing habits is generated without his knowledge. On the merchant's site it may not be guaranteed that only authorized staff may access personal data. A breach of confidentiality might also occur if data about shopping habits of customers or customer groups are published on the merchant's web site. During an unguarded transmission, data are readable by everyone with access to the transmission media.
- **Threats to integrity:** On the customer's computer the integrity of the data to be transmitted could be put at risk by malicious software such as Trojan horses or malicious applets. On the merchant's web site the data presenting his merchandise may be compromised by an attacker. A breach of integrity also occurs if an unauthorized user changes another user's data. During the transmission data may also be manipulated. Even though data are protected

against transmission errors on the lower layers of the TCP/IP-protocol stack, deliberate damage could take place since it is possible for an attacker to easily recompute the checksums of the protocol.

- **Threats to availability:** The availability of the customer's computer may be disturbed by malicious software like computer viruses or instable application systems. Attackers aiming to harm a certain merchant might try to attack his web site to set it out of order (e.g. through IP bombing). To guarantee certain availability of transmission services the Internet was built redundantly. However, current implementations of TCP/IP allow attackers to disturb the operation of computers or parts of the network.
- **Threats to accountability:** Either customer or merchant could forge a false identity towards each other. There are well known techniques like IP spoofing that produces a fake IP address of the sender's computer.

2.2 Existing Techniques for Securing Online Transaction Process

A number of factors must be taken in consideration to do e-transaction using credit/debit cards in a secure and trusted way; these factors are illustrated in [5]. Solutions must address many of these factors, without compromising the "ease of use by customers" feature.

The most popular techniques used for e-transactions are:

1. Secure Electronic Transaction (SET).
2. Secure Sockets Layer (SSL).
3. Payer Authentication Service like visa (3D Secure).

Each one of these techniques has its advantages and its drawbacks. These techniques are discussed in details in the following resources [5], [6], [7] and [8]. In general the main drawbacks of these existing techniques can be summarized as follows:

- Vulnerable to many attacks
- Expected benefits outnumbered the real ones
- Less safety and high cost
- Requires special software to be present in the customer's machine

3 Security Using Biometrics

With the rapid evolution of e-commerce, online shopping, electronic banking, and the increased concern on the privacy and security of the information stored in various databases, personal authentication and identification have become very important topics in security researches. A perfect authentication system will necessarily have a biometric component to make use of the power of human body. In this paper, only one of the biometric traits – viz., cardholder's fingerprints, has been employed to authenticate him.

3.1 Fingerprints as a Biometric Verification System

The biological properties of fingerprint formation are well understood and fingerprints have been used for identification and verification purposes for many years.

Fingerprints have been broadly used for identification of criminals by the various forensic departments around the world, and in biometric systems such as civilian and commercial identification devices since the beginning of 20th century.

Among all biometric traits, fingerprints prove to possess one of the highest levels of reliability [9]. Fingerprints are believed to be unique for individuals, and even across fingers of that same individual. It has been proved that fingerprints varied even in identical twins who has similar DNA structures. Moreover the use of fingerprint-based biometric systems offer positive verification with a very high degree of confidence, and small solid state fingerprint sensors may be easily embedded in the devices like keyboard and mouse. Owing to all these reasons, fingerprint-based authentication is becoming more popular in a number of civilian and commercial applications [8].

3.2 Multimodal Biometrics

Multimodal biometrics is an upcoming security mechanism which involves using more than one physiological or behavioral characteristic for enrollment, verification or identification. There is a great need for **multimodal biometrics** as most biometric systems used in real applications are unimodal, which means they rely on only one area of identification. Unimodal systems face high false acceptance rate and false rejection rate, limited discrimination capability, and lack of permanence. The limitations of unimodal biometric systems can be overcome by using multimodal biometrics where two or more sources are used to validate identity. Multimodal systems are more reliable because of the use of many independent biometrics that meet very high performance requirements They also effectively deter spoofing because it is near impossible to spoof multiple biometric traits and the system can request the user to present random traits that only a live person can do. In spite of the huge complexity involved in building such a system, it is preferred in applications like E-payment security which demand high reliability. As a preliminary effort, this paper discusses the use of unimodal paradigm where fingerprints are used currently for security. However the use of multimodal systems is promising in enhancing e-payment security.

3.3 Vulnerabilities in Biometric Authentication Systems

Even though, biometrics solves many security problems, it is suspicious towards many attacks. As people leave their fingerprint everywhere, anyone can hack the fingerprint. To avoid this, some principal features in fingerprint are extracted and stored as templates. Even this system faces some attacks as shown in Figure 1.

The attacks are

1. A bogus biometric trait like an artificial finger or its synthetic image may be presented at the sensor.
2. Illegitimately seized data may be resubmitted to the system.
3. A Trojan horse program can supply pre-determined feature sets instead of the genuine feature extractor.
4. Legitimate feature sets may be replaced with synthetic/cooked-up feature sets.

5. The matcher may be cheated by a Trojan horse program that always yields high scores thereby compromising the system security.
6. The templates stored in the database may be altered or deleted, or new templates may be populated into the database.
7. The data moving in the communication channel across different modules of the system may be changes, and
8. The final decision may be superseded.

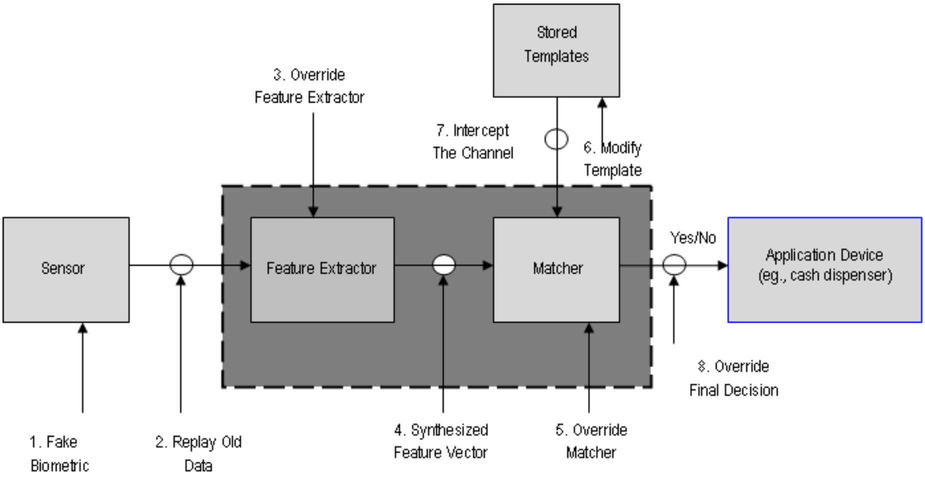


Fig. 1. Vulnerabilities in the existing biometric system

Besides these attacks, there are some specific attacks that take place in biometrics under suspicious case. They are

i. **Circumvention:** A hacker may gain access to the system protected by biometrics and peruse sensitive data.
ii. **Repudiation:** A legitimate user may access the facilities offered by an application and later claims that an intruder had circumvented the system.
iii. **Covert acquisition**: An impostor may secretly obtain the raw biometric data of an user to access the system.
iv. **Collusion:** An individual with wide super-user privileges (such as an administrator) may deliberately modify system parameters to permit incursions by an intruder.
v. **Coercion:** An impostor may force a legitimate user (e.g., at gunpoint) to grant him access to the system.

The important short comings of existing pattern are storing the fingerprint and pin number apparently as such and storing a copy directly in database. To overcome this limitation, in the current proposal, a sequence of procedures and certain modifications are incorporated to achieve more security. The main transformations which have been added in this proposal are the generation of B-PASS card with the given fingerprint and pin number so as to survive all attacks and a modified procedure for e-payments.

4 Proposed System

In this paper, a new online transaction system which is based on keeping the private cardholder's information more secure and which cannot be known by the merchant web site or even by the acquirer is proposed. The new system will give the e-customers the confidence to shop online without any worries about the Internet frauds because their information are hidden and cannot be extracted by any party except for the trusted card issuer party. Any customer who likes to benefit from the online shopping has to use an online shopping card or any credit card that is valid to be used over the Internet, and so he has to ask for one of these special cards. In the proposed system, the e-customer will be given this card beside a storage disk that contains software. This software will be used to authenticate the cardholder and to complete the online transaction by generating a special image called Biometric- Based Personal Authentication using Steganography Scheme (B-PASS); this image will contain the cardholder's necessary information to complete the transaction. The core idea of B-PASS scheme is that instead of using finger print and pin number for transaction which leads to many possible attacks, he must use the B-PASS card, fingerprint and pin number for transaction. If any of the hackers, acquire the fingerprint or pin number, he/she is not possible to do the transaction as they need the B-PASS card image for transaction. Transaction is possible only when all the three components are available. The sample B-PASS card image is shown in Figure 2.

Fig. 2. Sample B-PASS Card Image

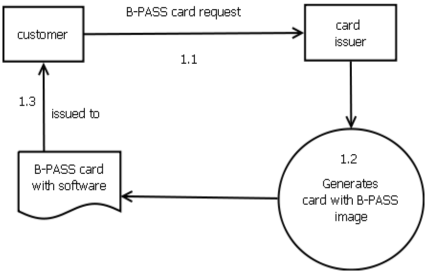


Fig. 3. Registration Phase in the Proposed System

The new system embodies two techniques that can be summarized as follows: the first technique uses a fingerprint verification technique as a biometric personal authentication system. The second technique is a fragile steganography algorithm in which the data hiding technique first encrypts the card information, and it embeds the encrypted data and the fingerprint in the cover image in a secure way in which the information embedded cannot be extracted unless a secret key is used.

4.1 Stages of the Proposed System

A B-PASS Card Request (Registration)

1.1 The customer requests for a B-PASS card for making online transactions
1.2 The B-PASS image generation processes.
1.2.1 The customer gives his fingerprint and personal code (pin).
1.2.2 The pin is embedded into the fingerprint resulting in stego fingerprint.
1.2.3 The card issuer provides the card number and the validity date.
1.2.4 The card details are encrypted using RSA.
1.2.5 The encrypted card sensitive data and the stego fingerprint are binarized and concatenated to generate the new B-PASS image.
1.2.6 Finally the generated B-PASS image is uploaded into the software.
1.3 The credit/debit card and fingerprint scanner with the software loaded into it, are finally given to the customer.

Figures 3 and 4 describe the complete functioning of the registration phase starting from a new customer requesting the card issuer for a B-Pass card till he is issued one.

B Using B-PASS in E-Transaction

Using B-PASS in the e-payment stage consists of the following steps.

(a) On completing the purchasing order, the cardholder will be asked to provide his transaction card number and expiration date as an ordinary way of e-payment or he will be asked to upload a valid B-PASS image to complete the transaction. To generate a valid B-PASS, the cardholder should use the software given to him by the issuer.

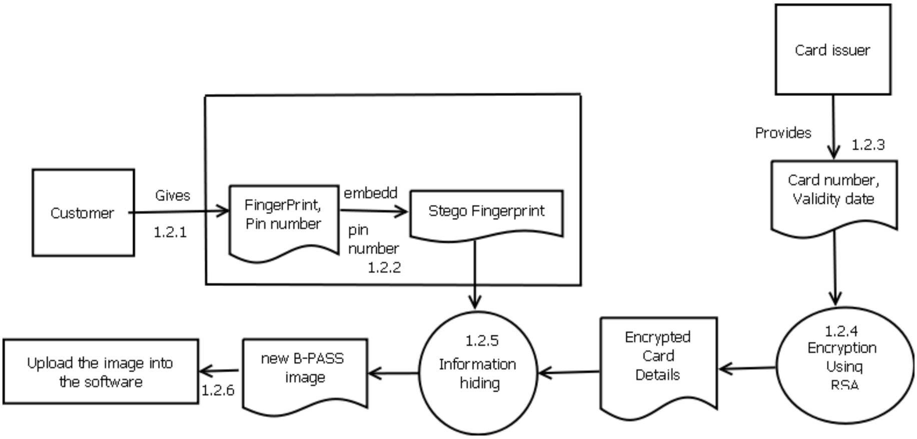


Fig. 4. The Flow Diagram of the Registration Phase

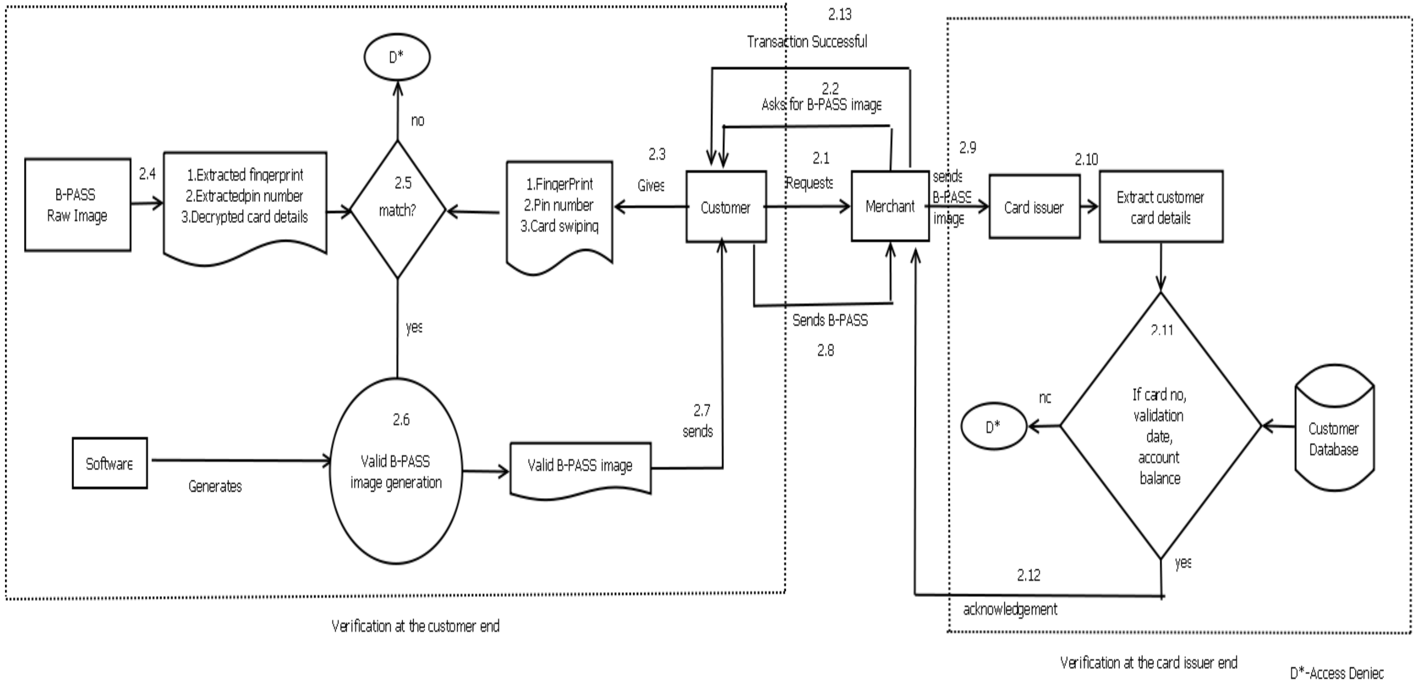


Fig. 5. The Functional Diagram of the Verification Phase

(b) The software will verify the cardholder by asking him to provide his fingerprint to complete the authentication process. Also he supplies his PIN which is embedded into the fingerprint image. The authentication process is done by comparing the cardholder's fingerprint image containing PIN with the raw B-PASS provided by the issuer during registration phase. If the verification process succeeds, the cardholder will be asked to enter the transaction amount, and he generates a valid B-PASS.

(c) The software will add a validation tag to the generated image; this tag contains a binary stream of the serial number and the amount of the transaction. The validation tag contains the transaction amount and transaction serial number.

(d) The generated valid B-PASS image has to be uploaded to the merchant's web site who is waiting for response.

C B-PASS Verification Stage

2.1 The customer requests the merchant for online transaction.

2.2 The merchant asks the customer for the generated B-PASS image

2.3 The customer provides his fingerprint, pin number and card details by swiping his card.

2.4 The original fingerprint, pin number and the decrypted card details are extracted from the B-PASS raw image.

2.5 The extracted details in 2.3 and the inputs given by the customer in 2.4 are compared for a match.

2.6 On finding an exact match, a valid B-PASS image with transaction number and transaction amount tagged with it, is generated by the software.

2.7 The generated valid B-PASS image is received by the customer.

2.8 The customer sends this B-PASS image to the merchant.

2.9 The merchant validates the B-PASS image with the card issuer.

2.10 The card issuer extracts the customer details from the received B-PASS image.

2.11 The extracted customer details are compared with the database record of that customer for the details like card validity date and account balance.

2.12 Once the user is validated, an acknowledgement is sent to the merchant.

2.13 Finally the merchant acknowledges the customer about successful transaction. Figure 5 shows the complete functional diagram of the verification phase. This involves processing at two ends namely – customer end and the card issuer end.

5 Security Considerations and Solutions

The new proposed e-transaction system considers and solves the following security issues:

The B-PASS image can only be used by its owner. This is because the authentication process is done before permitting the user to use the software that generates the B-PASS. This means losing the B-PASS or the software may not be a serious threat and no hacking is possible merely with them.

Giving the B-PASS image to the merchant is not a problem like in the ordinary online transaction system because the merchant has no idea about the real transaction card number or its expiration date. Accordingly, any hacking attempts on his servers will be useless for the hacker since these images are one-time-valid images.

Further trying to make copies of the B-PASS image will not be useful because each B-PASS must contain a unique validation tag which cannot be generated by anyone other than the cardholder.

The new system meets the different kinds of security objectives as follows:

Confidentiality: The new system prevents any unauthorized person from using others' online transaction cards or B-PASS by providing the authentication step that uses one of the cardholder's fingerprints.

Integrity: Any attack done on the B-PASS or any distortion on it caused by the transmission will not affect the information integrity because the whole transaction will be canceled.

Availability: Since the cardholder carries the storage media that contains the software, he can complete his online purchasing any time without any limits.

Accountability: The authentication step gives the merchants a level of confidence that the customer is genuine. The customer is not sending any sensitive information like PIN, card number etc., apparently over the public Internet. On the other hand, the customers need not worry even if the merchant is not genuine because the merchant can do nothing with the B-PASS images.

The advantages of the proposed system and its comparison with the existing solutions like SSL, SET and 3-D Secure [11], [12], [13] are compared in the Tables 1 and 2.

Table 1. Proposed System vs. SSL, SET and 3D-secure

	SSL	*SET*	*3D-secure*	*Proposed System*
Confidentiality	Yes	Yes	Yes	Yes
Integrity	Yes	Yes	Yes	Yes
Authentication	Optional for client	Yes	Yes	Yes
Anonymity	No	Yes	Yes	Yes
Merchant does not store sensitive data	No	Yes	Yes	Yes
Non-repudiation	No	Yes but complicated	Yes	Yes
Non-revocation	No	No	No	Yes
Protection of sensitive data on departure	No	No	No	Yes
Protection of sensitive data on arrival	No	No	No	Yes
Payment responsibility	Merchant or bank	Cardholder or bank	Cardholder	Card-holder
Complexity	Low	High	High	Low
Cost	Low	High	High	Low

Table 2. Advantages of the proposed B-PASS system

Customer side security	Transfer security	Card issuers security
-Protection of cardholder sensitive information at client side in B-PASS card -Combines fingerprint, PIN and validation tag for securing each transaction	-Authentication of cardholder, merchant and issuer bank -Integrity by embedding PIN inside the fingerprint image -Confidentiality -Simplicity of the payment process -Reduction of the logistics and the cost of implementation (compared to SET and 3D-secure)	-Encryption of trace files and logs -Encryption of bank database -Management of server access -Respecting the PCI standard

6 Conclusion

The proposed online shopping system is based on using a special image that acts as an electronic shopping card called B-PASS card. This special image contains customers fingerprint and contains the shopping card information. These data are hidden in the B-PASS card using a robust steganography algorithm like Spread Spectrum method. The proposed online transaction system keeps the private cardholder's information secure. It is not known by the merchant's web site or even by the acquirer, which will give the e-customers the confidence to shop online without any worries about the Internet frauds because their information are hidden and cannot be extracted by any party other than the trusted card issuer party. The new online transaction system considers a number of main online transaction security solutions like confidentiality, integrity, availability and accountability. Hence the proposed model can be employed for any e-payment field applications.

Acknowledgements. This paper is based upon work supported by the All India Council for Technical Education - Research Promotion Scheme under Grant No. 20/AICTE/RIFD/RPS(POLICY-II)65/2012-13.

References

1. Gartner, Inc.: The Evolution of e-Business Security Requirements. VeriSign. Inc White Paper, 1–52 (2001)
2. Winch, G., Joyce, P.: Exploring the dynamics of building, and losing, consumer trust in B2C e-Business. International Journal of Retail and Distribution Management 34(7), 541–555 (2006)

3. Jain, A.K., Ross, A., Pankanti, S.: Biometrics: A Tool for Information Security. IEEE Transactions on Information Forensics and Security 1(2), 125–144 (2006)
4. Knorr, K., Röhrig, S.: Security of Electronic Business Applications: Structure and Quantification. In: Bauknecht, K., Madria, S.K., Pernul, G. (eds.) EC-Web 2000. LNCS, vol. 1875, pp. 25–37. Springer, Heidelberg (2000)
5. Jewson, R.: E-payments: Credit Cards on the Internet. Aconite White paper, 1–33 (2001)
6. Anderson, R.: Security Engineering - A Guide to Building Dependable Distributed Systems. WILEY Computer Publishing (2001)
7. Wolrath, C.: Secure Electronic Transaction: a market survey and a test implementation of SET technology. Master Thesis, UPPSALA University (1998)
8. Cappelli, R., Maio, D., Maltoni, D., Wayman, J.L., Jain, A.K.: Performance evaluation of fingerprint verification systems. IEEE Trans. Pattern Anal. Mach. Intell. 28(1), 3–18 (2006)
9. Ross, A.: Information Fusion in fingerprint Authentication. PhD Thesis, Michigan State University (2003)
10. Maniam, B., Naranjo, L., Subramaniam, G.: E-Commerce Best Practices: How to Achieve an Environment of Trust and Security. International Journal of Innovation, Management and Technology 3(4), 396–401 (2012)
11. Houmani, H., Mejri, M.: Formal analysis of SET and NSL protocols using the interpretation functions-based method. Journal of Computer Networks and Communications 12, 36–48 (2012)
12. Visa International, 3-D Secure Introduction, Visa International Service Association (2002)
13. GPayments, VISA 3-D Secure vs. MasterCard SPA, `http://www.gpayments.com`

A Novel Cloud Based NIDPS for Smartphones

Vijay Anand Pandian and T. Gireesh Kumar

TIFAC CORE in Cyber Security, Amrita Vishwa Vidyapeetham University,
Coimbatore, Tamilnadu, India
{vijayanandrp,gireeshkumart}@gmail.com

Abstract. Internet usage via smartphones becomes higher which catches the attention of malicious cyber attackers to target their cyber threats over smart phones. Data being sent out from phone carries as packets contains lots of private and confidential information about the user. This paper proposes and evaluates an enhanced security model and architecture to provide an Internet security as a service for the smartphone users. It uses a cloud environment, includes VPN Server for the secure communication and network-based IDS and IPS provided with different machine learning detectors to analyze the real-time network traffic and serves as a user-friendly firewall. We also propose a D-S Evidence theory of information fusion to enhance the accuracy of detecting the malicious activity. Empirical result suggests that the proposed framework is effective in detecting the anomaly network activity by malicious smartphones and intruders.

Keywords: Smartphone security, Intrusion Detection, Intrusion Prevention, Cloud Computing, Machine Learning.

1 Introduction

In this modern era, the rising importance of electronic gadgets (i.e., smartphones) which became an integral part of business, providing connectivity with the Internet – brings many challenges to secure this device from being a victim of cybercrime. By the end of 2013 around 7.1 billion individuals around the world would have Internet access [1]. ABI Research forecasted that 1.4 billion smartphones will be in use before the end of 2013, with a growing proportion of those smartphones enabled for the Internet access [2]. People from different age groups use smartphones mainly for Internet-related activities [3]. This smart mobile technology is rapidly gaining popularity and cyber attackers (hackers and crackers) are among its biggest fans. A key driver for the growth of smart mobile technology is the rapid growth of business solutions into smart gadgets. The complexity of managing these smartphones outside the wall becomes the challenging issue.

Smartphone users use different Internet Service Providers (ISP) and Wi-Fi for accessing the Internet. The data packets during Internet communication comprise user private and confidential information i.e., Personally Identifiable Information (PII) that includes user name, national identification number (e.g., SSN, IMEI, IMSI), mobile

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 473–484, 2014.

phone numbers, personal texts (e.g., SMS, MMS, social network messages), birthday and digital identity (e.g., E-Mail address, Online account ID and password).

In recent years, the mobile threats have been raised abruptly [4]. Malwares perform harmful actions such as stealing data (PII) and uses the user device as a victim to enter the computer network to perform cybercrimes over their targets. Potentially unwanted applications (e.g., Spyware, Adware and Trackware) are undesirable and intrusive, considered as the major mobile threats which can bring the user privacy and security at risks. McAfee Labs reported about SMS-stealing banking malware, fraudulent dating apps, data-stealing apps and more ransomware samples were found in 2013 than in all previous periods combined [5]. Web based threats are re-directs, spam e-mail, spam URLS, push and popup flash Ads, auto untrusted Apps download, phishing, cookies stealing and session hijacking.

Some eminent kinds of attacks to which smartphones can be subjected to are: push attack - the attacker uses the smartphones as a botnet or zombies to perform DDoS attack on any targeted victim over network, pull attack - the attacker captures the WEP encrypted packets via rouge access points from smartphones that includes captured E-Mail, logins, passwords, etc., and crash attack - the attacker performing DoS attack over smartphones. Another major concern is that revealing the user public IP address will give a cyber criminal an opportunity to launch malicious attacks, used to retrieve user physical address and information about user profile and a Wi-Fi intruder/hacker can use user public IP address to download illegal materials and target their cyber attacks.

Necessities of security solutions for smartphones are in great demand. As smartphones have limited power, memory, processing speed and lack of super user mode makes difficult to provide the real-time protection. Majority of the smartphones do not have pre-installed security software. More anti-virus applications provide host end protection, are available in market. But it takes time to update new malware signatures and needs to be updated regularly. Application level security is provided by sandbox mechanism in smartphones. Another issue is that all applications during installation do not give the details about their server address and port number. Very less application provides network and Internet security for the smartphones.

There is a need for a single secure point for the smartphone users to provide safety, security and privacy in accessing network and it should restrain the users from variety of network based mobile threats. Moreover in IP data world, the attacks can be committed against smartphones can originate from two primary vectors, outside the mobile network (i.e., public Internet, private network and other operator's network) and within the mobile network. Designing IDS and IPS along with firewall for the smartphone is a big challenge.

There are mainly two types of intrusion detection: signature-based and anomaly-based. The deployment of IDS can be done in two forms: Host-based IDS (HIDS) protects the system by auditing and monitoring the events and logs. Network-based IDS (NIDS) deals with monitoring and analyzing the network traffic. IDS and IPS can be implemented locally or on a remote server. But implementing locally requires high amount of computation, more memory to process and consumes more power on the device, lowers the user experience. IDS and IPS on a remote server hosted in cloud has lot of positives.

Cloud computing is a hot technology for virtualization and it is very popular due to its converged infrastructure, shared services, flexibility and scalability. Drawback of implementing HIDS in cloud needs more number of features to be extracted from the host smartphone at regular intervals and perform analysis of extracted features and then make decisions on the state of the device and also it requires Internet access at all times for synchronizing the feature vectors. Network-based IDS and IPS (NIDPS) in the cloud are more advantageous as it drops attacks, logs packets and terminates sessions. NIDPS can monitor more number of smartphones and provides protection in real time. Firewall hosted in cloud is pre-configured with white and black lists, rules and policies which can prevent the users from malicious websites and unknown network attacks. A combination of IDS, IPS and firewall provides a better network security.

In this paper we present an idea for secure smartphone communication, called SSC framework for analyzing the network traffic and provides an Internet security. In SSC framework, the cloud includes VPN server, NIDPS and a firewall. VPN provides a private tunnel which hides the user IP address, prevents man-in-the-middle attacks and secure the communications across the Internet. Various machine learning algorithms are implemented in NIDS which act as detectors. Results from these detectors are fused using the D-S Evidence theory for improving the accuracy rates to identify the malicious activity over smartphone network traffic flows. Each detector runs as a separate thread module to classify the normal and abnormal activity. Thread model in our framework performs parallel computing in real-time so that it can consume very less time in detecting malicious activity.

The rest of the paper is organized as follows: Section 2 provides an overview of related work in the area of smartphone intrusion detection and cloud based security. Section 3 explains the proposed architecture. Section 4 describes evaluations and experiments performed, while conclusions and future works are given in Section 5.

2 Related Works

Thomas et al. [6], presented a proxy running in a cloud environment that controls the access for mobile device, developed based on the Role Based Access Control (RBAC) model.

Zhizhong et al. [7], proposed a framework of client-server architecture where the mobile agent continuously extracted various features and send to the server to detect anomaly using anomaly detectors. They used multiple distributed servers with different machine learning as a detector for analyzing the feature vector and D-S Evidence theory of information fusion is used to fuse the results of detectors, also proposed a cycle-based statistical approach to find anomaly activity. This distributed detectors idea and D-S Evidence theory of result fusion method are adopted in our work to compute the joint mass which can be effectively used in the parallel and distributed computing systems.

Jianxin et al. [8], presented security challenges in green cloud computing, and designed CyberGuarder for virtualization security assurance and proposed a virtual network security service with trust management and access control.

J. Wright et al. [9], presented a survey about cyber security and mobile threats for smartphones and discussed the security model of the android and iOS.

Abdul et al. [10], presented a survey on the secure mobile cloud computing and identified the potential problem and suggested an idea for security issues in data and application security frameworks provided with a taxonomy of the state of the art.

Caner et al. [11], evaluated a WallDroid, which used virtualized application specific firewalls managed by the cloud, VPN technologies with the Point to Point Tunneling Protocol (PPTP) and the Android Cloud to Device Messaging Framework (C2DM), aimed to track the android applications and their network activity to find their reputation.

Saman et al. [12], proposed a Secloud that emulates a smartphone device in a cloud environment and keeps it synchronized by continuously passing the device inputs and network connections to the cloud and allows Secloud to perform a resource-intensive security analysis on the emulated replica such as monitoring the network traffic, system events and behavior by using IDS.

Cloud security achieves the abnormal detection of the network software behavior through the reticular large client, then gets the latest information of Trojans and malicious programs in the Internet and pushes the information to the server to analyze and finally server distributes the solution of viruses and Trojans to each client [13].

Miao et al. [14], proposed a cloud communication firewall to filter harassing phone calls and spam messages, collect, process and share information dynamically to other clients. This cloud methodology for monitoring the network activity and a firewall idea exactly suits our problem domain.

There has been considerable amount of research about anomaly detection in computing system network traffic [15, 16]. These include statistical-based approaches [17-19], data-mining based methods [20], and machine learning based techniques [21, 22] and the combination of D-S Evidence Theory [23].

From the survey reports, more and more complex detection methods and architecture were applied into smartphone intrusion detection that includes emulated devices, virtual replica and so on. But most of the ideas and works were interesting. Cloud computing, VPN technology, a cloud firewall, result fusion of distributed detectors with different machine learning, dynamic information sharing and various types of IDS methodology are the key ideas that motivated us to propose a solution for our problem domain.

We aim to establish a network and Internet security services that overcome the smartphones computation, memory and storage problems for implementing the real time IDS, IPS and a firewall. Our solution is to provide secure communication for billion of smartphones across the untrusted networks and the Internet by enforcing tunnelling protocols, NIDPS for real time intrusion detection and prevention and a user customizable firewall. Key reasons for choosing a cloud computing for its converged infrastructure i.e., group collaboration, virtualization, centralized management, scalability, elasticity, flexibility, low costs, increased data reliability, unlimited storage capacity, data recovery, security and device independence [24]. Cloud can handle large pool of users and provides services in real time. Network-based threats mainly Denial-of-Service (DoS) from the remote server, ranks top in the security vulnerabilities of android [25].

3 SSC Architecture Overview

The top view of SSC framework is illustrated as Fig. 1. The VPN clients connect to the VPN server via GPRS, EDGE, 3G or Wi-Fi. Fig. 2 presents the NIDPS high level architecture.

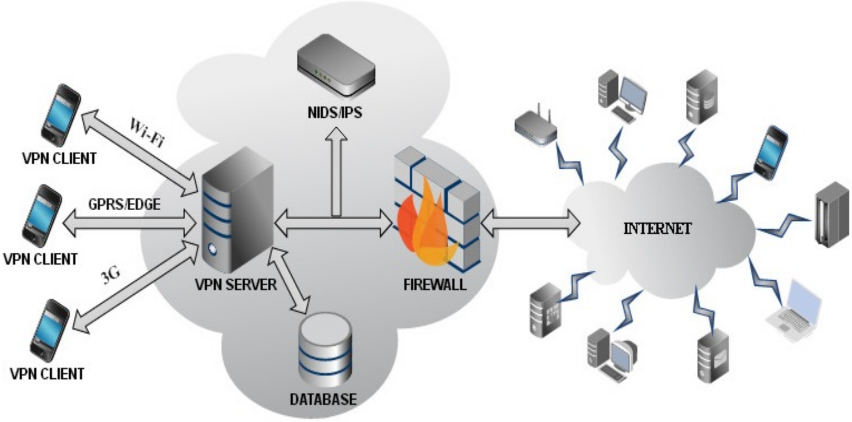


Fig. 1. Secure Smartphone Communication (SSC) High Level Architecture

The concept of cloud computing comprises of three service models: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS).

In our proposed model, Iaas model (private cloud) used to support operations, including storage, hardware, servers and networking components. VPN server, NIDPS server and firewall server (default gateway) are virtually hosted and interconnected in cloud. VPN client i.e., mobile agent uses the cloud service as an end-user. Once VPN client is connected, mobile network traffic flows through the cloud and real-time protection is given to the user. By hosting multiple of VPN servers, NIDPS servers and firewall servers and by sharing the common storage server in cloud at different locations in the world, it is possible to give the real time Internet security and protection for the large number of smartphones from network based threats.

3.1 VPN Client

VPN client is a mobile application. Initially on the first run, the client registers their username and password with VPN server site. After verification of the user, the configuration and connection is setup between VPN client and VPN server. When the setup phase is over, VPN server acts as default gateway for accessing any public network (Internet). Each data packets from the client device is encrypted and transmitted to VPN server. IPSec protocol is used for encryption and decryption and also provides authentication, integrity and confidentiality. IPSec encrypts and decrypts the

IP packets with a key shared between VPN client and VPN server by Internet Key Exchange (IKE). Layer 2 tunneling protocol (L2TP) provides a delivery service for ISP and it relies on IPSec protocol. Point-to-point Protocol (PPP) provides a connection between two nodes in the data link layer. A user can monitor the data usage and also the frequent domain or websites visited. Another important feature is that user can block any IP address or websites by adding it in the user blacklist (custom) maintained by VPN server. This can be further implemented as a parental control system in the mobile phones along with network monitor that provides a statistical report. Usage of the tunnelling protocol prevents the smartphone from man-in-the-middle attack and eavesdropping.

3.2 VPN Server

The communication is established between the client and the server by the session key exchange and the authentication by verifying username and password. The packets received from the client are decrypted and the requests are forwarded to the respective server that the client intends to connect. Here the source address is replaced by the VPN server address for preventing the disclosure of the client identity. On the reply of the requested server, the destination address is replaced with the client address, encrypted and forwarded to the client device. The VPN server maintains the user records for authentication such as username and password. It keeps device information such as OS name and IMEI. It logs the user information such as websites, IP addresses and time stamps during the communication in the secured database managed in the storage server. Managing and storing logs plays the key role in detecting anomalies in the network and reconnaissance scans from potential break-ins. The client-customized firewall rules and policy will be stored in the VPN server. Each packet from the client is checked against the rules created by the client and drops the packets in the event of a violation in the VPN server itself. This reduces the firewall server load that discussed in the next section. On detection of any anomalous traffic or activity reported by NIDPS, an alert will be pushed back to the client device from VPN server. This ensures the awareness to the clients about the malicious activity so that the clients can add new rules in their custom firewall.

3.3 Firewall

Firewall is placed between router and VPN server in cloud. Firewall act as a default gateway for the whole cloud setup. Firewall analyzes the network traffic that are incoming and outgoing. It decides whether to accept or block the communication based on applied rule set. Creation of rule sets and configuration of policies are done by the administrator in the firewall server. Maintenance of white and black lists avoids the web based attacks such as phishing attack, malicious URL and malicious content websites and addresses. These lists are updated manually and stored in the common database. Firewall acts as main component for blocking the unwanted communication and intruders in the network.

3.4 NIDPS Server

NIDPS Server is placed between VPN server and the firewall. NIDS consists of several systems that are implemented with various machine learning algorithms for detecting the normal and abnormal activity. Main work of NIDS is to monitor the network traffic and to extract the feature vectors. The extracted features are given to the different detector system for classification and the results from each detector are fused together before sending to IPS. After the correlation of the result, alert is sent to the administrator. IPS enables the administrator to take manual or automated response in real time against intrusions by adding the new firewall rules.

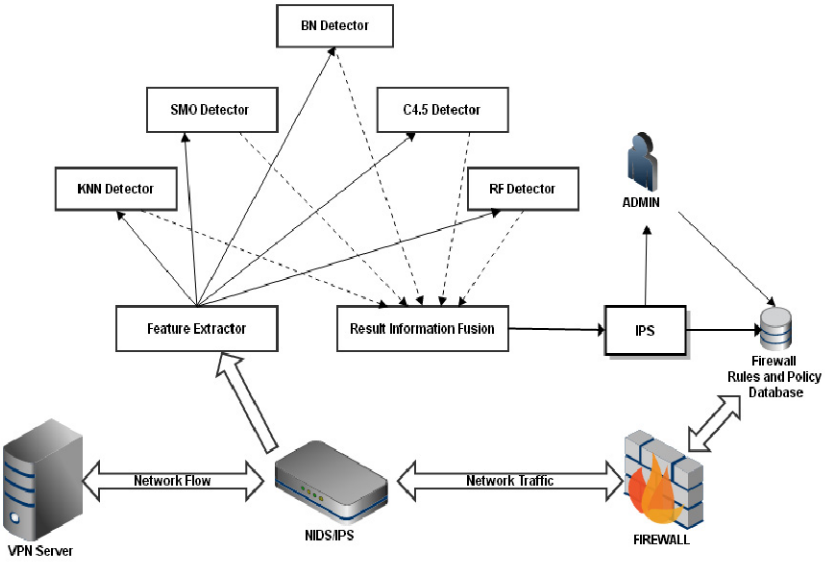


Fig. 2. Proposed NIDPS High Level Architecture

Feature extractor component extracts maximum number of features from the network traffic and sends required feature vector to the each detector. Multiple machine learning and classifying algorithms used in the detector module detects different malicious and suspicious activity such as spamming, DDoS, phishing, botnets, malicious upload and download over the network traffic. Each detector in the SSC framework is a separate thread. All threads executes in parallel to produce the results. NIDPS logs the network activity for auditing. Various attack patterns that are detected by NIDS are stored in the storage server.

4 Experiments

In order to evaluate our proposed model, we performed several experiments. The research question is described in the section 4.1. Section 4.2 describes the creation of the dataset for experiments. In Section 4.3 describes the tools and results of our experiment.

4.1 Research Question

Our experiments aim at resolving the following inquiries:

1. Which classification algorithm is most appropriate for IDS?
2. How many numbers of extracted features and feature selection method yield the most detection results?
3. How to improve the Accuracy and TPR using D-S evidence theory?

In the information fusion system [7], $\Theta = \{N, A\}$, where N = normal, A = abnormal. The combination rule can be described as:

$$K = [m_1(\{N\}) * m_2(\{A\}) + m_1(\{A\}) * m_2(\{N\})] \tag{1}$$

$$m(\{N\}) = \frac{m_1(\{N\}) * m_2(\{N\}) + m_1(\{N\}) * m_2(\{\Theta\}) + m_1(\{\Theta\}) * m_2(\{N\})}{1-K} \tag{2}$$

$$m(\{A\}) = \frac{m_1(\{A\}) * m_2(\{A\}) + m_1(\{A\}) * m_2(\{\Theta\}) + m_1(\{\Theta\}) * m_2(\{A\})}{1-K} \tag{3}$$

To achieve the efficiency of the various detection algorithms and feature selection schemes, we employed the following 3 standard metrics: *Accuracy*, which measures the proportion of absolutely correctly classified instances, either positive of negative; *True Positive Rate (TPR)*, which is the proportion of positive instances classified correctly; and *False Positive Rate (FPR)*, which is the proportion of negative instances misclassified. Detection results are classified into 4 categories: *TP* is number of positive instances classified correctly; *FP* is the number of negative instances misclassified; *FN* is the number of positive instances misclassified; and *TN* is the number of negative instances classified correctly. The formula for the 3 metric as follows,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$TPR = \frac{TP}{TP + FN} \tag{5}$$

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

4.2 Creating Dataset

1. *Collection of Normal Dataset*

We created a Wi-Fi hotspot that connected with 3 smartphones. Wireshark, an open source packet analyzer is placed between hotspot machine and default gateway machine. Pcap file is created consisting of captured packets of smartphones. We used scapy, a powerful interactive packet manipulation python script to collect the features from the pcap file and made a normal dataset for our experiment.

2. *Collection of Abnormal Dataset*

We used android malicious applications to perform network-based suspicious activity (i.e., HTTP upload & HTTP download of larger files, frequent use of ping command). These packets are captured using wireshark. Scapy is used for feature extraction.

3. *Features Extracted*

We collected 30 features from the pcap file for our experiment. Features extracted are source and destination IP, duration, protocol type, service, source and destination bytes, payload, flag, server send and received error rate, destination host send and received error rate, server and destination host count.

4.3 Scheme of Experiments and Results

We used weka, an open source tool for machine learning algorithms. The dataset for this experiment includes all trusted and malicious applications feature vectors of the same device. The training dataset contained 80% of the instances, and the testing dataset contained the rest 20% instances. We tested with different classification algorithms and their accuracy is given in the Table 1 which answers our 1st research question.

Table 1. Accuracy of Each Classification Algorithms

Classification Algorithm	**Accuracy**
BN (Bayesian Network)	0.972
NB (Native Bayes)	0.904
SMO (Sequential Minimal Optimization)	0.973
J48 (C4.5 Decision Tree)	0.996
RF (Random Forest)	0.997
RT (Random Tree)	0.995
DT (Decision Table)	0.990

We evaluate the effect of training set size and degree of anomaly behavioral detection accuracy. Fig. 3 presents the average accuracy of four feature selection methods and different number of top features gives solution to our 2nd research question. The result indicates that the accuracy maintains a high rate when the number of top features are higher than 10 and falls rapidly when the number top features lesser than 10. Feature selection methods used in weka tools are InfoGain, GainRatio, ReliefF and SymmetricalUncert.

The result shows that the total detection time used for result fusion is little higher than the single detector. Table 2 presents the Accuracy, TPR and FPR for each combination of algorithms, and the following conclusions can be drawn: (1) the fusion result is same as the better detector when the results of two detectors are changed (2) the detection performance is better in most cases of result fusion. The table 2 answers our 3rd research question and also shows that D-S Evidence theory plays an important role in distributed and parallel. Another major advantage of our proposed model is that our VPN client application does not requires larger computation and memory.

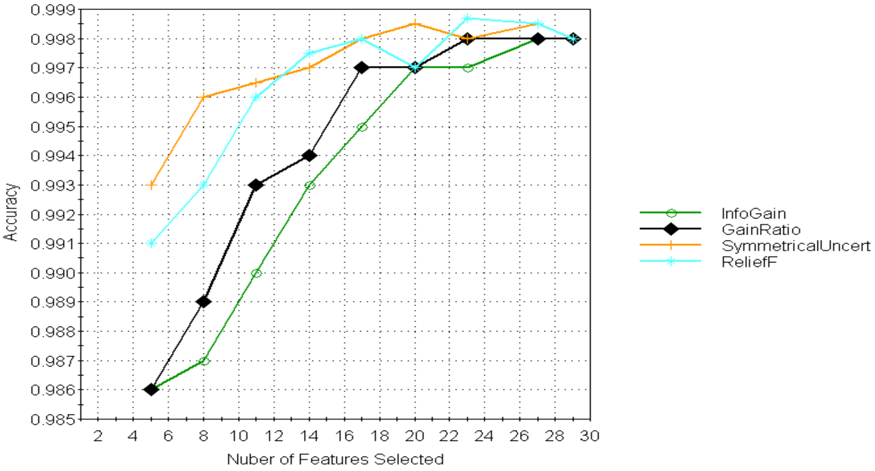


Fig. 3. Average Accuracy of each feature selection method and number of top features

It consumes very less battery power as the analyses are performed on the remote servers hosted in cloud.

Table 2. Accuracy, TPR and FPR of fusing each combination of algorithms

ALGORITHM	ACCURACY	TPR	FPR
BN	0.5431	0.7694	0.0976
SMO	0.5675	0.5967	0.0087
J48	0.6673	0.4541	0.0256
RF	0.6433	0.7781	0.0190
BN, SMO	0.6997	0.8286	0.4001
BN, J48	0.7129	0.8399	0.0063
BN, RF	0.9764	0.8914	0.2006
SMO, J48	0.9551	0.8174	0.1445
J48, RF	0.9009	0.7088	0.0008
BN, SMO, J48	0.8338	0.6645	0.0057
SMO, J48, RF	0.8854	0.6873	0.0012
BN, J48, RF	0.9276	0.8926	0.1223
BN, SMO, J48, RF	0.9568	0.9481	0.0011

5 Conclusion

This paper proposes and evaluates an enhanced security model and architecture to provide an Internet security for millions of smartphone users. VPN Server used in cloud provides secure communication. In NIDPS server, each detector corresponds to a unique classification algorithm that distinguishes normal and abnormal feature vector. The D-S Evidence theory of information fusion is used for better results.

5.1 Future Work

The proposed idea can be further enhanced to make a solution for Mobile Device Management (MDM) and Bring Your Own Device (BYOD) related problems. HIDS module can be added in cloud by synchronizing the logs and system events of the smartphone for every interval to the detector servers. Handling billions of smartphone users to provide network security is also a big issue. The design and implementation cost, time and scope for this proposed model should be evaluated further.

References

1. Global Internet Usage, http://en.wikipedia.org/wiki/Global_Internet_usage
2. ABI Research, https://www.abiresearch.com/press/45-million-windows-phone-and-20-million-blackberry
3. Office for National statistics, Internet Access - Households and Individuals (2013), http://www.ons.gov.uk/ons/dcp171778_322713.pdf
4. Symantec Intelligence Report (November 2013), http://www.symantec.com/connect/blogs/symantec-intelligence-report-november-2013
5. McAfee Threats Report: Quarter (2013), http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q2-2013.pdf
6. Thomas, R., Christoph, R.: Enhancing Mobile Device Security by Security Level Integration in a Cloud Proxy, in ThinkMind. In: The Third International Conference on Cloud Computing, GRIDs, and Virtualization, Nice, France, pp. 159–168 (2012)
7. Zhizhong, W., Xuehai, Z., Jun, X.: A Result Fusion based Distributed Anomaly Detection System for Android Smartphones. Journal of Networks 8(2) (2013)
8. Jianxin, L., Bo, L., Tianyu, W., Jinpeng, H., et al.: CyberGuarder: A Virtualization Security Assurance Architecture for Green Cloud Computing. Future Generation Computer Systems 28(2), 379–390 (2012)
9. Wright, J., Dawson Jr., M.E., Omar, M.: Cyber Security and Mobile Threats: The Need For Antivirus Applications For Smart Phones. Journal of Information Systems Technology & Planning 5(14), 40–60 (2012)
10. Abdul, N.K., Mat Kiah, M.L., Samee, U.K., Sajjad, A.M.: Towards secure mobile cloud computing: A survey. Future Generation Computer Systems, 1278–1299 (2013)
11. Caner, K., Todd, B., Karl, A.: WallDroid: Cloud Assisted Virtualized Application Specific Firewalls for the Android OS. In: Proceedings of IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (2012)
12. Zonouz, S., Amir, H., Berthier, R., Borisov, N., Sanders, W.: Secloud: A cloud-based comprehensive and lightweight security solution for smartphonesl. Elsevier on Computers & Security 37, 215–227 (2013)
13. Xu, H., Yuan, J.: Research on Cloud Monitoring Oriented to Mobile Terminal. Computer Science 39, 55–58 (2012)
14. Miao, C., Qinsheng, H., Fangfang, J., Qiao, D.: Research of Cloud Security Communication Firewall Based on Android Platform. In: Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (2013)
15. Patcha, A.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks, 3448–3470 (2007)

16. Chandola, V., Banerjee, A., Kumar, V.: Anomaly Detection: A Survey. ACM Computing Surveys, 15–58 (2009)
17. D'Alconzo, A., Coluccia, A., Ricciato, F., Romirer-Maierhofer, P.: A Distribution-Based Approach to Anomaly Detection and Application to 3G Mobile Traffic in Global Telecommunications Conference (2009)
18. Raimondo, M., Tajvidi, N.: A peaks over threshold model for change point detection by wavelets. Statistica Sinica 14 (2004)
19. Wang, H., Zhang, D., Shin, K.: Statistical analysis of network traffic for adaptive faults detection. IEEE Trans. Neural Networks 16(5), 1053–1063 (2005)
20. Prashanth, G., Prashanth, V., Jayashree, P., Srinivasan, N.: Using random forests for network-based anomaly detection. In: IEEE ICSCN 2008, Chennai, India, pp. 93–96 (2008)
21. Shon, T., Kim, Y., Lee, C., Moon, J.: A machine learning framework for network anomaly detection using SVM and GA. In: IEEE Workshop on Information Assurance and Security. US Military Academy, West Point (2005)
22. Li, Y., Guo, L.: An efficient network anomaly detection scheme based on TCM-KNN algorithm and data reduction mechanism. In: IEEE Workshop on Information Assurance and Security. US Military Academy, West Point (2007)
23. Sentz, K., Ferson, S.: Combination of Evidence in Dempster-Shafer theory in SAND, pp. 0835 (2002)
24. Cloud Computing, `http://en.wikipedia.org/wiki/Cloud_computing`
25. Google, Android: Security Vulnerabilities, `http://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/Google-Android.html`

Preserving Database Privacy in Cloud Computing

Saed Alrabaee[1], Khaled Khateeb[2], and Fawaz A. Khasawneh[3]

[1] Computer Security Laboratory, Concordia University, Montreal, Canada
saed.ece@gmail.com
[2] Jordan University of Science and Technology, Irbid, Jordan
Kikhateeb@just.edu.jo
[3] ETS - University of Quebec, Montreal, Canada
fawaz.khasawneh.1@ens.etsmtl.ca

Abstract. Due to the rapid advances in the networking technologies and the continued growth of the internet have triggered a new trend of cloud computing towards outsourcing data to the cloud service providers. The ever cheaper and most powerful database as a service (DaaS) computing paradigm that enables organizations to minimize their operational cost in a way that they no longer need to purchase infrastructure and hire human resources. In this paper, we need to present a secure model that provides data security at cloud and anonymization of data in a way that satisfies e-differential privacy.

Keywords: SAAS, Cloud Computing, Data Privacy.

1 Introduction

Due to the rapid advances in the networking technologies and the continued growth of the internet have triggered a new trend of cloud computing towards outsourcing data to the cloud service providers. The ever cheaper and most powerful database as a service (DaaS) computing paradigm that enables organizations to minimize their operational cost in a way that they no longer need to purchase infrastructure and hire human resources. By outsourcing the workload to the cloud service provider, organizations could use unlimited computing resources by paying affordable service charges without investing in software, hardware and operational overheads [1].

Despite all these benefits, the major obstacle towards the large adoption of cloud computing paradigm is the data security and privacy concerns. In cloud computing (DaaS), the data owner outsources his private, sensitive data and querying services to the cloud provider, which is basically an untrusted server. Data owner needs protection of his data from cloud and the querying clients. While data gives superb opportunities to querying clients in relation with data mining tasks but opens the privacy issues as well. On the other hand client also seeks privacy of his queries to cloud and data owner. In this paper, we need to present a secure model that provides data security at cloud and anonymization of data in a way that satisfies e-differential privacy [2].

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 485–495, 2014.

1.1 Motivations

Existing frameworks that provide the security and privacy in the cloud computing environment are based on two kind of framework. In a trusted server framework, there has to be one additional trusted server placed outside cloud and that works between client and cloud for secrecy. In the other framework, client needs to perform encryption, decryption, distance and other calculation to avoid trusted server.

Moreover, existing database-As-a-Service (DaaS) models are unable to support advanced queries such as aggregation while maintaining the secrecy of data simultaneously [3]. Aggregate queries permit one to retrieve succinct information such as counters from such a database, since they can cover many data items while returning a small result. The shared storage area motives our work. For example, various users access the storage for different purposes like the database relates to the hospital, it could be accessed by doctors and researcher as well. Hence, for some types of information and some classes of cloud computing users, privacy and confidentiality rights, onuses, and status may change when a user discloses information to a cloud provider.

1.2 Contributions

Theft of sensitive private data and trade of information in the open market for profit is significant problem. Online database management systems (DBMSs) are a lucrative target for malicious users either for gaining sensitive information or even just for the fun of penetrating organizations network for sensitive data as they often contain huge volume of sensitive information. When individual users or enterprises store their sensitive data in a DBMS today, they must trust that the server hardware and software are uncompromised, that the data center itself is physically protected, and assume that the system and database administrators (DBAs) are trustworthy. But nowadays, more and more organizations both small and large are undertaking internet to capture business and they do not have that budget to setup their own datacenter so they are getting more biased towards the cloud technology for service. But as sensitive data of the organizations will be stored in a third party location and under their full control so it is a matter of concern from the organizations perspective to ensure the proper privacy of their data from illegitimate access.

In this paper, to address this privacy concern of database a new framework has been designed and implemented to fit organizations long pending demanding storing corporate database in a third party location in a secure manner and also ensuring secure access control on the data to legitimate users. In this paper, we assumed that the key server on the client end is trusted and the only entrusted part in our setup is the cloud storage server which is hosted in an entrusted third party location. Figure 1, shows the general cloud computing.

The rest of this paper is organized in the following manner: Section 2, cloud computing architecture review. Section 3, formally defines the problem. Section 4, elaborates the different modules of our framework. Section 5, Experimental results and section 6. Section 7, conclusion and future directions are presented.

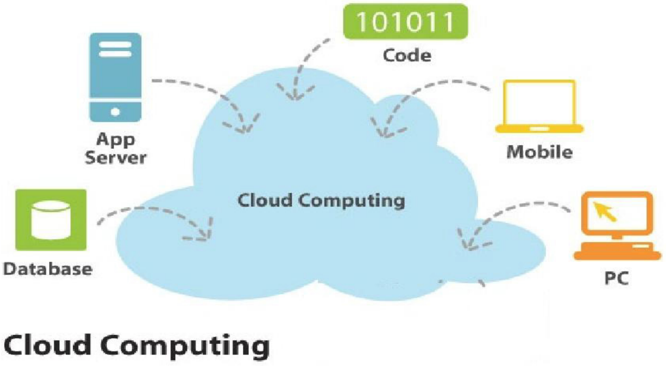


Fig. 1. A general cloud computing

2 Related Work

Over the past few years, researchers have developed cryptographic tools for encrypted database that is stored in cloud storage as well as for searching keywords over encrypted text [1, 2] and some researchers have proposed using these tools to process SQL queries on encrypted data [3, 4]. Basically, the closest works related to our framework are [5, 6]. In [5], they use trusted cloud between the storage cloud and the user.

In our framework, we have a trusted key server instead of Trusted Cloud which is located either on the data owner server or on a different server, away from the cloud. It is cheap to have only a server instead of having trusted cloud, more flexibility and no restriction over the space for storing the keys. In the paper a secure channel was used as SSL/TLS. Basically SSL/TLS has strong capabilities offered for secure channel such as handshake protocol, MAC computation, PRF function for master secret and key material. In [6] the researchers use new encryption scheme which is called Onion encryption. Hence, we took the idea of layers and the idea of joining two different tables.

We provide a literature review of cloud computing features in Section A followed by a description of using cloud as storage in Section B.

2.1 Cloud Computing Architecture

Cloud computing, which dynamically provides reliable services over the Internet, is one of the most emerging technologies in current world. Recently, many academic and industrial organizations have started investigating and developing technologies and infrastructure for cloud computing. There presentative cloud platforms include Amazon Elastic Compute Cloud (EC2), Google App Engine, and Microsoft Live Mesh. Mainly 3 (three) types of services we can get from cloud service provider as SaaS, PaaS and IaaS.

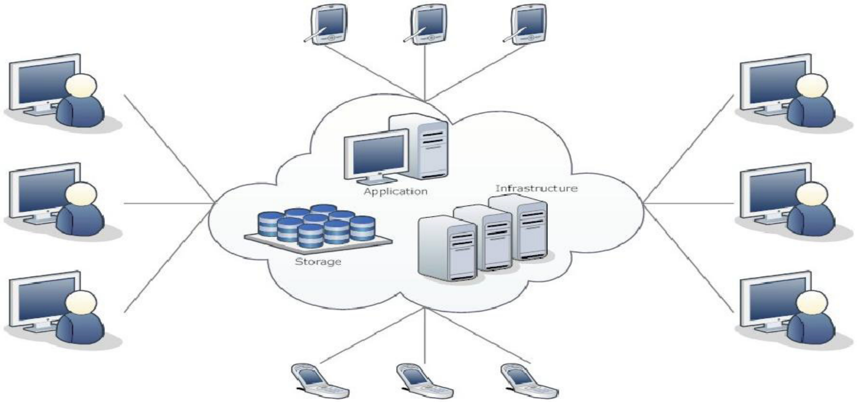


Fig. 2. Storage Cloud Computing Architecture

2.2 SaaS in Cloud Computing

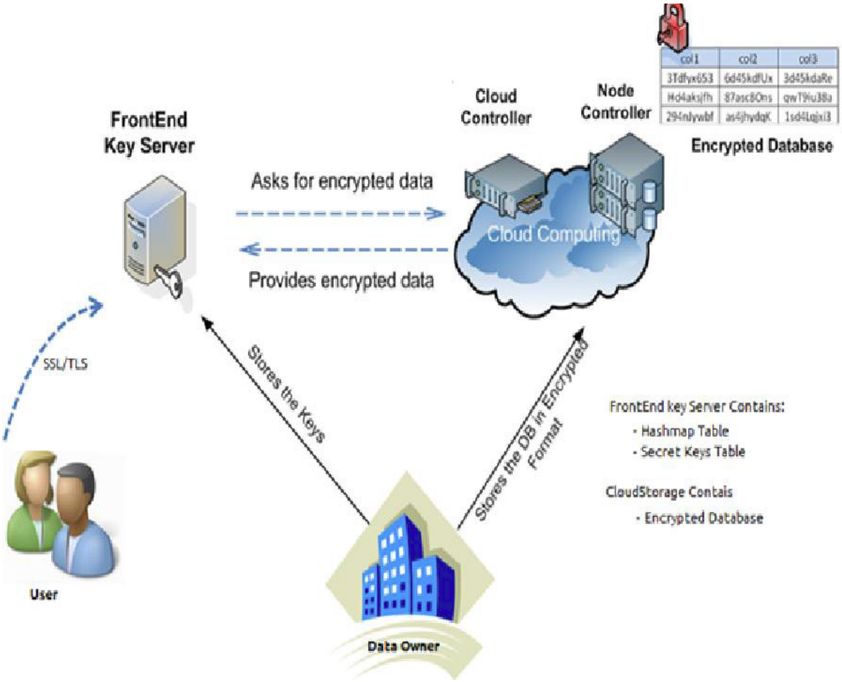
Cloud computing involves highly available massive compute and storage platforms offering a wide range of services. One of the most popular and basic cloud computing services is storage-as-a-service (SAAS). It provides companies with affordable storage, professional maintenance and adjustable space. On one hand, due to above-mentioned benefits, companies are excited by the public debut of SAAS. On the other hand, companies are reticent about adopting SAAS. One of the major concerns is the privacy as the cloud service is generally provided by a third party.

3 Proposed Approach

In our framework, we tried to address the following challenges regarding how to preserve the privacy of the database on the cloud:

- Challenge 1- how to protect outsourced data from theft by hackers or malware infiltrating the cloud server?
- Challenge 2 - how to protect outsourced data from abuse by the cloud server?
- Challenge 3 - how to realize content-level fine grained access control for users

The general idea of our proposed method, depicted in Fig. 3, can be summarized in five phases: (1) Preliminary Stage: implementation of the cloud using UEC (Ubuntu Enterprise Cloud) in VMware (2) Key server (trusted): is deployed for users' access control and legitimate access over cloud database. At the end of second stage the connectivity between the user and the key server has also been encrypted using secure SSL/TLS. (3) Encryption of the whole database based on dynamic key generation from the key server. (4) Migration of organizations database to cloud. (5)
Access control mechanism for user accessing data stored in cloud.



col1	col2	col3
3Tdfyx653	6d45kdfUx	3d45kdaRe
Hd4aksjfh	87asc8Ons	qwT9iu38a
294nlywbf	as4jhydqK	1sd4Lqjxi3

Fig. 3. Proposed cloud setup

3.1 Cloud Setup

To implement a cloud computing environment, we first need to define what type of services our cloud will provide and what resources we need to support those services. In our case, we consider that an HR department of an organization wants to store a database on a cloud. This database consists of several tables, among which there are tables with more sensitive information. In practice, large databases require certain resource capacity and well trained personal to manage it. Having on mind this, our HR department would need a storage location which can provide all of the above requirements. Therefore, our cloud should provide storage services. These services will be provided to the users of the HR department in a remotely accessible fashion. Several deployment models exist: Public cloud, Community cloud, Hybrid cloud and Private cloud. For the purposes of our framework, we simulate a Public cloud which is considered as an untrusted storage, managed by a cloud provider.

To set up our cloud we need two separate machines -one will be the cloud controller and the other will be the node controller. One of these physical machines will have the cloud controller installed as a virtual machine on VMware and also will accommodate the user host. The other physical machine is used only for the node controller and it has CPU Virtualization enabled which is a requirement to have a successful setup. The cloud controller consists of storage controller, cluster controller, Walrus and the cloud controller itself. The cloud controller will provide the front end to the entire cloud infrastructure. The node controller is used to manage the instances that can be run on the node. For the installation of the cloud and the node controller we are using the Ubuntu Enterprise Cloud (UEC), which is free software and includes Eucalyptus which is a cloud platform.

First, we need to install the cloud controller using our UEC software. Because of hardware limitations, we use VMware to install the cloud controller as a virtual machine. During the installation, some important steps are to give a name of our cluster, to give a range of IP addresses which later will be given to each instance that tries to connect to the cloud controller. The next step is to connect the machine that has the cloud controller with the machine that will be used to install the node controller. For simplicity in this paper, we use a crossover cable to connect physically the two machines and we configure them to be in the same subnet. Then we can start installing the node controller while the two machines are connected. This will ensure that the installation of the node will automatically detect the existing cloud controller and will associate with it. During this installation, the node gets registered with the cloud and public SSH keys are being exchanged, as well as, the services are configured and published. Having set up the main components of the cloud, we need to do several more configurations to provide full operability.

The next step is to make sure each user of the cloud can obtain credentials from the cloud. There is an admin user who can manage all the user accounts. The credentials need to be downloaded from the cloud which can be reached through a web browser from the node or from any other location in the network through the URL: //cloud-ip-address:8443/. After that, certain images are proposed that can be installed on the cloud. These images are stored on the Walrus controller and are used to create instances on the cloud. We can install one of them, which is enough for this simulation, so we have installed the image Ubuntu 9.10 Karmic Koala. Later on, the instances can be managed by different tools such as: Elastic fox or Hybrid Fox, and command line euca2ools. But before running any instances, we need to make a key pair that is used to login as a root. Store the client's database.

It can be seen that the cloud provider can have full access to the data that the client wants to store on the cloud. So the data confidentiality and integrity can be compromised. Having set up this, we can now start our instance which is basically a virtual machine run on the node. On this instance we install a Linux-Apache-MySQL-PHP (LAMP) server which will be used to protect this data, the client has to enforce encryption of the database before it to be stored on the cloud. This leads to complications for providing queried information to the users, but we have achieved to implement a way which is both functional and secure.

3.2 Encryption Scheme

Basically the cloud storage is considered a very active part as storage area where it provides a good space for storage but the main concern is the privacy issue. The Encryption ensures the data stored is confidential and its privacy is preserved from the untrusted cloud or even from any attacker who tries to steal some useful data. The encryption in our framework is achieved by using different layers of encryption. The algorithm for the encryption is AES and to strengthen the security titles of the columns are hashed by MD5.

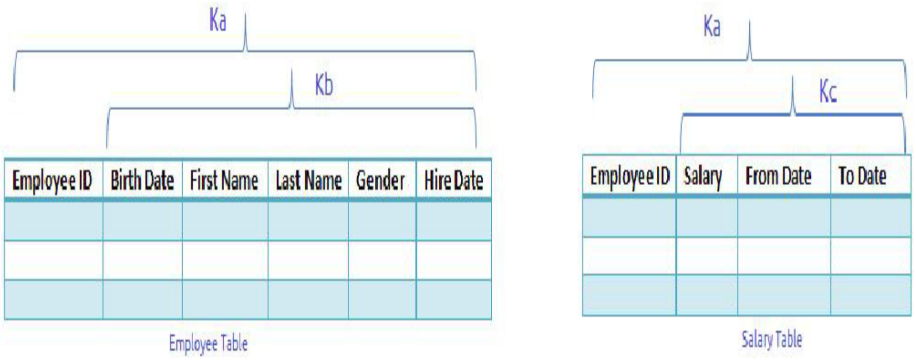


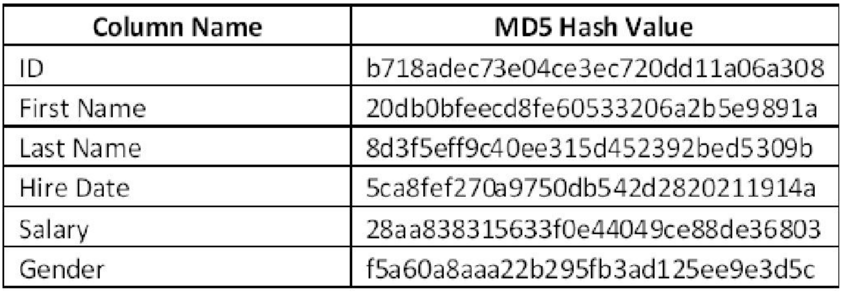
Fig. 4. Encrypted Table in Cloud

1. *Advanced Encryption Standard (AES):* It is symmetric-key encryption adopted by National Institute of Standards and Technology (NIST). Our choice for AES is based on the following characteristics: it is a simple design, a high speed algorithm, with low memory costs. Due to the fact the key security feature is the size of the key, we choose AES 128 as key length.

Encryption Layers: In our framework, we have two layers of encryption - the outset layer for both tables were encrypted with the same key but the inner columns - with different keys as shown on Fig. 4. In the above figure, even if the intruder has the key Kath at means the intruder can access the employee id column only because the other columns are encrypted by either Kc or Kb. So by using layers, we provide more confidentiality. However, if two columns indifferent tables are to be joined, they need to be encrypted with the same key. Hence, Ka is used for such purposes.

2. *Encrypted Query using MD5:* MD5 is cryptographic hash function that produces a 128bit hash value. Due to the fact the query privacy has become an important issue in the past few years because of confidentiality problems with the client queries, we decided to use the MD5 to enforce security of the queries. The clients, in some cases, require the secrecy for his query. To achieve this, we used MD5 to hash the values of the queries.

Table 1. Hashing Table



Column Name	MD5 Hash Value
ID	b718adec73e04ce3ec720dd11a06a308
First Name	20db0bfeecd8fe60533206a2b5e9891a
Last Name	8d3f5eff9c40ee315d452392bed5309b
Hire Date	5ca8fef270a9750db542d2820211914a
Salary	28aa838315633f0e44049ce88de36803
Gender	f5a60a8aaa22b295fb3ad125ee9e3d5c

The above table represents the MD5 value for some titles so in cases where the client or user needs to search for first name - let's say "Ali"- so the trusted key server transfers the first name to MD5 hash value and then sends the query through the network in the following manner:

- mysql> SELECT * FROM Table1 Where20db0bfeecd8fe60533206a2b5e9891a ='6d1baa8615bb02a4c779949127b612d4.

  Where '20db0bfeecd8fe60533206a2b5e9891a' is an encryption of the first name and'6d1baa8615bb02a4c779949127b612d4' - is the following:
- Ali is encrypted first by Kb so the value will be '7a9b46ab6d983a85dd4 d9a1aa64a3945' and then this value is encrypted by Ka so the value will be'6d1baa8615bb02a4c779949127b612d4'.

Moreover, if an intruder captures the clients' both the query and the result, he will never get any knowledge by combining both because the intruder cannot understand the hashed values so he will not know what the client is looking for.

3. *Key Management*

Having solved the problem of what scheme is going to be used to encrypt the database, another challenge arises which is finding a way of managing the keys used to encrypt and decrypt the database. This important issue has to solve the problem of what keys will be provided to what users. To implement this, first we need to distinguish the different types of users which is achieved based on the access control mechanism. The other step is to provide a trusted storage for the keys. For this purpose, we have a key server which is located in a totally different place than the cloud. We have implemented the Key server as a normal Ubuntu Server as a virtual machine on VMware. The key server is the connection between the user and the cloud. Whatever the user wants to search in the database, the query will first pass through the Key server and then transferred to the cloud. The Key server is the point where the query is encrypted and where the encrypted data is brought from the cloud and decrypted for the user.

1. *Key Generation:* To provide strong security, one of the main parts is to ensure the keys used for encryption and decryption are stored on a safe place and are managed in a way that is hard for any adversary to capture and make use of them. Therefore, we have decided that our keys will be stored on the key server and generated dynamically. This means, that in a certain period of time, the keys are being refreshed. So this ensures that even if an encryption key was captured, it cannot be used after that period.

Specifically, we use Symmetric keys, a public key that is the same for all of the departments and a private key that is different for each department. The public key is used to encrypt or decrypt the whole database. But then, the private key is used to encrypt and decrypt specific fields of the tables from the database. The relation between the department and the keys is stored on the key server in a table which maps the department with the assigned keys. This way, the key distribution happens completely transparent to the users – the users do not hold the keys and do not need to know the keys.

2. *Process Flow:* The data owner wants to store the data in an encrypted format on the cloud. And he wants to make sure the keys for encryption are stored on a separate location, which in our case is the key server.

   First, an operator user inserts data for the employees

- the operator browses the URL:

https://keyserver/CloudDBPrivacy/index.html which is stored on the Key server (the connection is secured via SSL/TLS)

- enters his credentials
- inserts the data and submits it

The data gets encrypted using the keys assigned to the operator and it is sent for storage to the cloud by the key server. Then an HR user wants to check who receives a certain amount of salary:

- HR logs in with his credentials
- The Key server verifies the credentials
- HR queries the salaries
- The Key server encrypts the query and sends it to the cloud
- The cloud provides the queried data in an encrypted format to the key server. The salary is stored on a different table than the general information, so the cloud provides only the queried data from the specific table.
- The Key server uses the assigned keys to the HR in order to decrypt the data and provides it to the HR's interface.

4 Experimental Results

The following diagram outlines the network topology used for the implementation.

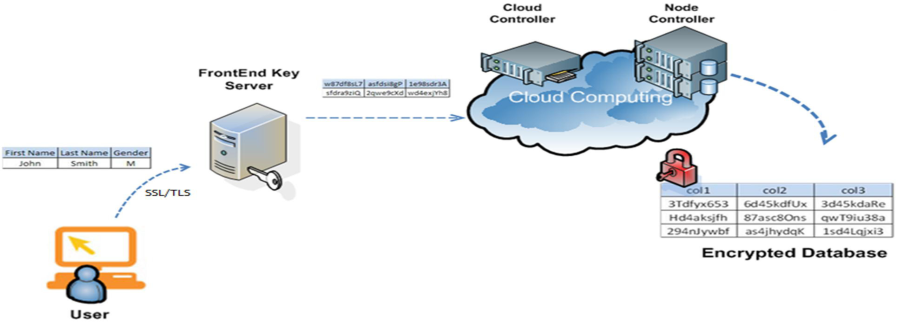
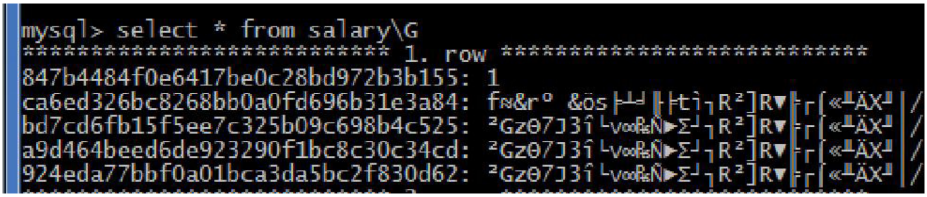

Cloud Controller
Node Controller
FrontEnd Key Server
w87df8sL7 | asfdsi8gP | 1e98sdr3A
sfdra9ziQ | 2qwe9cXd | wd4exjYh8
Cloud Computing
First Name | Last Name | Gender
John | Smith | M
SSL/TLS
col1 | col2 | col3
3Tdfyx653 | 6d45kdfUx | 3d45kdaRe
Hd4aksjfh | 87asc8Ons | qwT9iu38a
294nJywbf | as4jhydqK | 1sd4Lqjxi3
Encrypted Database
User

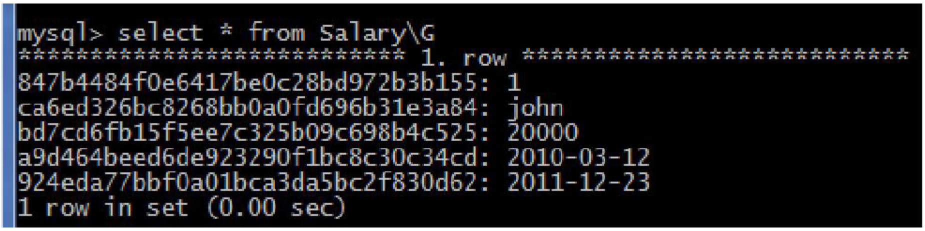
Fig. 5. Cloud network topology

Having simulated several cases, we have achieved the desired results.



```
mysql> select * from salary\G
*************************** 1. row ***************************
847b4484f0e6417be0c28bd972b3b155: 1
ca6ed326bc8268bb0a0fd696b31e3a84: f≈&r° &ös├┴┘║├╞tì┐R²]R▼╞┌[«╩ÄX╝│/
bd7cd6fb15f5ee7c325b09c698b4c525: ²Gzθ7J3î└v∞₧Ñ►Σ┘┐R²]R▼╞┌[«╩ÄX╝│/
a9d464beed6de923290f1bc8c30c34cd: ²Gzθ7J3î└v∞₧Ñ►Σ┘┐R²]R▼╞┌[«╩ÄX╝│/
924eda77bbf0a01bca3da5bc2f830d62: ²Gzθ7J3î└v∞₧Ñ►Σ┘┐R²]R▼╞┌[«╩ÄX╝│/
```

Fig. 6. Encrypted record of the database



```
mysql> select * from Salary\G
*************************** 1. row ***************************
847b4484f0e6417be0c28bd972b3b155: 1
ca6ed326bc8268bb0a0fd696b31e3a84: john
bd7cd6fb15f5ee7c325b09c698b4c525: 20000
a9d464beed6de923290f1bc8c30c34cd: 2010-03-12
924eda77bbf0a01bca3da5bc2f830d62: 2011-12-23
1 row in set (0.00 sec)
```

Fig. 7. Decrypted record of the database

In this paper, we addressed the problems as follows with solutions:

- To protect the outsourced data from theft by hackers we implemented encryption methodology so that without the proper secret key no adversary can decode the database. But it may have some performance degradation.
- Secondly, to protect the outsourced data from abuse by the cloud server we encrypt the database with secret keys before outsourcing the database to the cloud so that even the cloud administrator will not be able to guess the content of the stored database. Also the SQL queries from the client end are encrypted so that eavesdropping on the query itself will not be helpful to get the idea about the database.
- To address the content level fine grained access control we implemented another trusted key server which will create and store secret keys based on the user roles and will provide these keys to user on the fly while accessing the database in the cloud. Even users will not have any idea about the keys.

5 Conclusion

In this paper, we presented a new privacy preserving scheme in cloud storage environment, which meets the needs of the current industry. Preserving privacy of mission critical sensitive data of the enterprise world is of great importance. There are still some aspects that can be improved in our design. One weakness of our scheme is that there is no mechanism for checking the integrity of the stored data in the cloud. For now we just concentrated on the privacy of the data stored in a cloud, so the protection of data integrity is not carefully considered. We will try to mitigate the above integrity problem in our future work.

References

[1] Hu, H., Xu, J., Ren, C., Choi, B.: Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism. In: Proc. of the 27th IEEE International Conference on Data Engineering (ICDE 2011), Hannover, Germany (2011)

[2] Dwork, C.: Differential privacy. In: ICALP (2006)

[3] Thompson, B., Haber, S., Horne, W.G., Sander, T., Yao, D.: Privacy-Preserving Computation and Verification of Aggregate Queries on Outsourced Databases. In: Goldberg, I., Atallah, M.J. (eds.) PETS 2009. LNCS, vol. 5672, pp. 185–201. Springer, Heidelberg (2009)

[4] Chang, Y.-C., Mitzenmacher, M.: Privacy preserving keyword searches on remote encrypted data. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 442–455. Springer, Heidelberg (2005)

[5] Bugiel, S., Sadeghi, A.R., Schneider, T., Nürnberger, S.: Twin Clouds: An Architecture for Secure Cloud Computing (Extended Abstract). In: Workshop on Cryptography and Security in Clouds (CSC 2011) (March 2011)

[6] Popa, R.A., Zeldovich, N., Balakrishnan, H.: CryptDB: A Practical Encrypted Relational DBMS. Technical Report MIT-CSAIL-TR- 011-005. Computer Science and Artificial Intelligence Laboratory, Cambridge (2011)

A Survey on Wi-Fi Protocols:WPA and WPA2

Mahmoud Khasawneh, Izadeen Kajman, Rashed Alkhudaidy, and Anwar Althubyani

Faculty of Engineering and Computer Science, Concordia University, Montreal, Canada
{m_khasaw,i_kajman,r_alkhu,a_althu}@encs.concordia.ca

Abstract. Wireless Network (Wi-Fi) becomes extremely popular over the world on the lastly two decades. Nowadays, most people are using the wireless networks for online banking and shopping. It also allows people to share information and communicate with each other whenever and wherever they are. Moreover, it has the advantages of flexibility and freedom of mobility and enables smart phones and laptops to provide a rapid and easy access to information. All of that makes the protection of the wireless network highly demanded to be existed. Therefore, security should be applied on Wi-Fi networks to keep users' confidentiality and privacy. Hence, different protocols have been developed and applied. Nowadays, Wi-Fi Protected Access (WPA, and WPA2) protocols are considered as the most applied protocols in wireless networks over the world. In this paper, we discuss the advantages, vulnerability, and the weaknesses of both of these protocols. This paper ends up with some suggestions on how to improve the functionality of these protocols.

Keywords: Wi-Fi, Security, WPA, WPA2, Confidentiality.

1 Introduction

In the last years of the 20th century, Wi-Fi technology has accessed our houses without getting permission. In general, Wi-Fi is considered as one of the most commonly used and trusted technologies over the world. Wi-Fi networks are available everywhere, at school, at home, at hospitals, and at restaurants, etc... Therefore, users can access Wi-Fi networks anywhere and anytime with their laptops, PDAs, smart phones to share the pleasant moments with their friends.

The medium of all the data carried over Wi-Fi networks is open access i.e. the channels that carry the information exchanged in Wi-Fi networks are shared between the different users of different networks. However, this data should be securely exchanged between the users of Wi-Fi networks. Therefore, security concepts and issues have become a hot topic of research and investigation.

Starting in 1990, many wireless security protocols have been developed and adopted, but none of them could be considered as the best protocol ever because of the different security threats that daily arise with new vulnerabilities and problems to our data and applications.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 496–511, 2014.

Three main security protocols have been developed by the researchers which are: Wireless Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA, and WPA2) [1]. WEP was the first default encryption protocol introduced in the first IEEE 802.11 standard, aimed to make the wireless network as secure as the wired network. However due to its technical failures, it was not widely applied.

In order to solve the WEP problems of the cryptography method, Wi-Fi protected access (WPA) has been proposed [1]. Wi-Fi Protected Access2 (WPA2) is a new protocol that has been developed after both WEP and WPA failed to secure the communication over Wi-Fi networks [1]. WPA2, also known as IEEE 802.11i standard, is an improvement to the 802.11 standard which specify security mechanisms for wireless networks [2].

Through this paper, we address different issues of WPA and WPA2 protocols such as: protocols' architecture, security services provided threats, strengths and weakness [3].

2 Wi-Fi Protected Access (WPA)

Due to the fact that WEP is not secure enough, the IEEE 802.11 Task Group I (TGi) presented a new protocol which is the Wi-Fi Protected Access, widely known as WPA by improving WEP. WPA contains the Temporal Key Integrity Protocol (TKIP) [4]. There are two modes under which WPA functions: the first being Pre shared Key (PSK) and the other is Enterprise [5]. Typically, the Enterprise mode is more comprehensive in terms of security; it provides as it does not share any key or information but it is harder to set up than PSK. While RC4 Stream Cipher is used for encryption in WPA, there are three elements with which TKIP differs from WEP protocol which are: Michael, a message integrity code (MIC), a packet sequencing procedure, and a per packet key mixing [6]. Figure 1 shows the Flow of TKIP Processing.

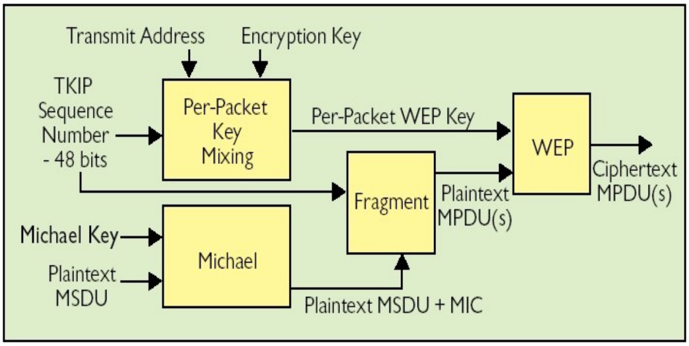


Fig. 1. Flow of TKIP Processing [4]

The main security features that are applied in WPA and different from WEP are as following:

2.1 WPA Encryption Process

- The Temporal Key Integrity Protocol (TKIP) is a protocol used for encryption and generating a new key for each packet. The size of each key is 128 bits.[2]
- For message integrity, there is an algorithm called "Michael". This algorithm is used to compute a Message Integrity Code (MIC) for TKIP, and (MIC) will be added to data to be sent.[3]
- In the encryption process, a new packet sequencing number will be processed to prove freshness of the packet that is being sent.
- For replay attack protection, TKIP offers two different data units which are: Medium Access Control Service Data Unit (MSDU) and Medium Access Control Protocol Data Unit (MPDU) [7].
- WPA uses RC4 for encryption process as WEP does but the main difference is that in TKIP the Base Kay and IV are hashed together before RC4 is being used. The result of hashing IV and the Base key will be used in RC4 with IV to generate a sequential key. The plaintext will be XORed with the sequential key and the result will be sent as a coded message [8]. The encryption algorithm is shown in figure2.

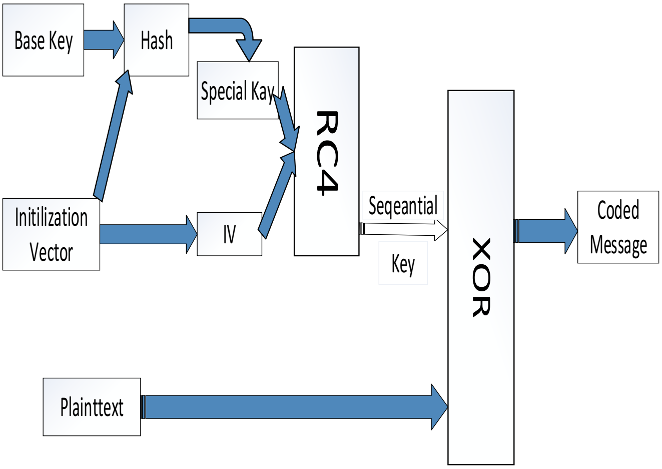


Fig. 2. WPA Encryption Algorithm (TKIP) [10]

2.2 WPA Authentication Mechanisms

In order to authenticate users and issue new keys that ensure a key management, TKIP utilizes the IEEE 802.1x standard, TKIP necessitates both a 64-bit key, that Michael uses, and a 128-bit key, that the aforementioned mixing function uses to receive a per packet key. In WPA there are two modes WPA Personal, WPA Enterprise and the authentication Mechanisms for each mode could be described as following:

- WPA Personal: It is also called WPA-PSK (Pre-Shared Key). This mode is usually used for home network or small office networks, and it does not use an authentication server. In this mode there is a key shard between the client and the access point (AP) and this key must be known to both sides for an association to be established. All wireless devices use 256 bits key to authenticate with their access point(s). It is extremely important that the shared key will never be transmitted between the client and the AP. By using the shard key the MIC and encryption key will be founded. MIC is in size of 64 bits and the encryption key size is 128 bits [4].
- WPA Enterprise: It is usually used for business network. No shred key is used in the authentication process; however an Extensible Authentication Protocol (EAP) is used. (EAP) offers two ways authentication. In this mode Remote Authentication Dial In User Service (RADIUS) server is obligatory and it delivers an excellent security for wireless network traffic [4].

3 Wi-Fi Protected Access (WPA2)

In 2004, the ratification of WPA2 is widely known as the second generation of WPA and it is recognized to be the most secure protocol used in wireless networks. This protocol uses the implementation of the 128-bits Advanced Encryption Standard (AES) block cipher algorithm for both authentication and encryption processes. In WPA2 there are two modes of authentication that could be used which are Pre-Shared Key and Enterprise. Instead of TKIP, WPA2 uses Pair wise Transient Key (PTK) for key generation. Instate of using Michael algorithm, WPA2 uses CCMP (Counter Mode CBC MAC Protocol) which applies block cipher Advanced Encryption Standard (AES) algorithm. In order to ensure integrity and provide accurate authentication, CCM (CBC-MAC) has been used in WPA2 [9].

3.1 WPA2 Encryption Process:

The encryption process, as shown in figure 3, could be done by applying the following steps:

- For each Medium access control Protocol Data Unit (MPDU) there is a packet number (PN) and this number will incremented for each next MPDU.

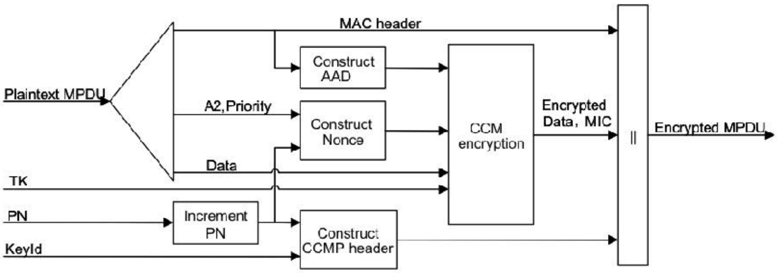


Fig. 3. CCMP Encryption Process [2]

- In the header of MPDU, there is something called Additional Authentication Data (AAD) and in this field the integrity delivered by CCMP is represented.
- To create the CCMP Nonce block the PN and, A2 (MPDU address 2) and Priority field of MPDU will be used. The Priority field has reserved value of zero.
- In addition the new PN with the key identifier together will be used to build the 64 bit CCMP header.
- The group of temporal key, AAD, nonce, and MPDU data are used to create the cipher text and MIC.
- Finally, the encryption of MPDU is obtained by combining the CCMP header, original MPDU header, encrypted data and MIC [2].

3.2 WPA2 Decryption Process

WPA2 does not use the XOR to decrypt the plaintext, and the decryption process will be done in the same steps. The steps for decryption, as shown in figure 4, are described as following:

- After the encrypted MPDU is received, the AAD and nonce values could be extracted from the encrypted MPDU.
- The header of the encrypted MPDU is used to build the AAD.
- To create the nonce value, the values of different fields of the header will be used which are the MPDU address 2 (A2), PN, and Priority fields.
- To recover the MPDU plaintext, temporal key, MIC, AAD, nonce and MPDU cipher text data are combined together. Moreover at this point the integrity of AAD and MPDU plaintext is confirmed.
- Finally, by combining MAC header of MPDU and decrypted MPDU plaintext data, the Plaintext of MPDU is decrypted See figure4 [2].

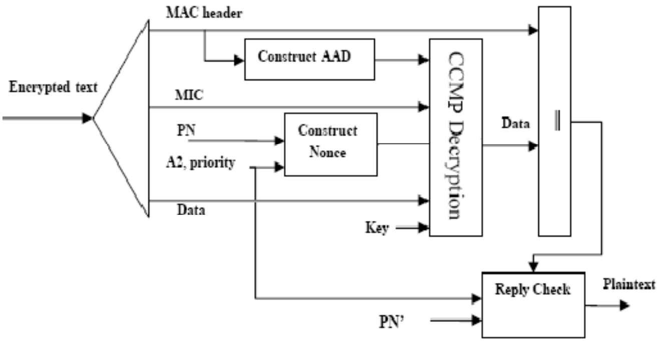


Fig. 4. CCMP Decryption Process [2]

3.3 Authentication Mechanisms

For authentication there are two types of key management systems, each of which utilize different means: an authentication server system or a pre-shared key system and once the keys are generated the authentication could be done same as it is used in WPA [4]. There are two types of key management systems which are described as following:

- A pre-shared key system is less comprehensive in terms of security than a system that uses an authentication server. However, despite such relatively incomprehensive security and the fact that complete implementation of the 802.11i protocol restrict any usage of pre-shared keys, small business and home users can still implement and utilize pre-shared keys with much ease[10].
- A system that generates keys through an authentication server is relatively hierarchical; what facilitate the creation of matching Pairwise Master Keys (PMK) at both supplicant and the authentication server sides is the 802.1x key generation protocols. Every time a device interacts with an AP, four types of 128-bit temporal keys, known as the Pairwise Transient Key (PTK), are generated: they are a data encryption key, a data integrity key, EAPOL-key encryption, and EAPOL-key integrity key [11]. In order not to only increase randomness but also relate the keys to its creator device, the key incorporates a random nonce and MAC addresses of the device. Then there are the four exchange ways called 4-way handshake between the AP and the authentication server, which identify and verify the key. Following the first step, which is the generation of a temporal keys and a pair of nonces, which the supplicant and authenticator have made, the supplicant verifies its knowledge of the PMK, and subsequently, the authenticator does so too. Finally, both devices have encryption turned on for unicast packets [10]. Also, it should be noted that 802.11i supports broadcast messages. In order to ensure efficiency, a Group Master Key (GMK) is created, and the GMK facilitates the generation of the

Group Encryption Key and Group Integrity Key that all participating g clients receive through a secure channel. Figure 4 illustrates how the protocols differ from each other. It is interesting to note that a hardware upgrade is necessary for 802.11i [12].

4 Comparison of Wi-Fi Protocols: WPA and WPA2

4.1 Data Integrity: WPA and WPA2

WPA uses Michael to verify the integrity of the message. In addition the Packet Sequencing is used to prevent replay attacks. On the other hand, WPA2 uses CCMP to provide integrity for both data and packet header [13]. A 48-bit sequence number that changes whenever there is a replacement of a MIC key prevents replay attacks; this sequence is what TKIP utilizes and labels as packet sequencing. The method mixes the aforementioned sequence number with the encryption key, encrypting the MIC and WEP ICV while detecting and removing packets that contain an out-of-sequence number. This section represents the Michael and Counter Mode with Cipher Block Chaining MAC Protocol (CCMP) protocols [7].

4.1.1 Michael or MIC

These MIC algorithms protect data integrity from the modification that could be caused by counterfeiting and forging [14]. Simply, a predefined algorithm and data both together calculate a tag value that the sender transmits with the key and the comparison between the sent value and the other value that the receiver calculates determine whether data integrity is intact or not [10]. In particular, Michael necessitates a new 64-bit key and represented as two 32-bit little Endian words (K_0 ,K_1). The functionality of MIC works as following:

- The length of total message is a multiple of 32-bits and to ensure that message will be a multiple of 32-bits add a message with the hexadecimal value 0x5A and enough zero pad.
- Dividing the message of a multiple of 32-bits into sequence of 32-bit words $(M_1 , M_2 \ldots\ldots M_n)$
- finally calculates the tag from the key and the message words by following format [10]:

```
$(L, R) \leftarrow (K_0 , K_1)$
do $i$ from 1 to $n$
$L \leftarrow L \; XOR \; M_i$
$(L,R) \leftarrow$ Swap $(L,R)$
return $(L,R)$ as the tag
```

- The verification step is done by matching the tag received with the message and the tag that achieved by computing the previous step.

- The time that an attacker needs to be able to build his/her MIC and not be detected is as following: if MIC is a S bits the average time will be after 2^{-S+1} packet [10].

4.1.2 Counter Mode with Cipher Block Chaining MAC Protocol (CCMP)

Michael was used in WPA for data integrity. Michael was developed to let existing wireless devices to overcome the several flaws of WEP. Michael may be implemented through software updates; it does not require hardware replacement of AP and STAs. However, it still depends on RC4 cryptographic algorithm so it is not a good solution for high assurance environments. Therefore, Counter Mode with Cipher Block Chaining MAC Protocol (CCMP) is developed and considered as a better solution [15]. However hardware upgrades are required for the wireless devices used.

CCMP relies on CCM which is a cipher mode of AES that is used to authenticate an encrypted block. In CCM, a 128-bit block size is ciphered. Cipher Block Chaining MAC (CBC-MAC) is used in CCM for both authentication and integrity protection. CCMP compromise the integrity of both the packet data and the MAC portion of the IEEE 802.11 header. In order to prevent replay attacks CCMP uses a nonce which is constructed by using a 48-bit packet number PN [16].

CCMP is considered as the best solution for both confidentiality and integrity. It uses the same cryptographic key for both confidentiality and integrity which reduces the complexity. CCMP provides integrity of both packet header and packet payload.

Figure 5 illustrates the integrity process of the packet's data and header. The process is done in 5 phases which are:

- Packet Number (PN) Increment: A 48- bits packet number used for each session is incremented. PN prevents replay attacks from occurring and ensures that the TK of each session lives more time than that of any possible STA-AP association.
- Nonce construction: A nonce is constructed by combining the packet number PN with the transmitter ad address (A2) as well as with the priority bits.
- CCMP Header Construction: a 48-bits Key ID used to identify the Temporal Key (TK) is joined with the incremented PN to form the CCMP Header.
- Additional Authentication Data (AAD) Construction: AAD is one of the important inputs to the CCM encryption module; it is either a 22-bytes or 28 bytes in length. It is constructed by using different fields of the MAC header such as the quality-of-service QoS parameter.
- CCM Encryption: it is considered the main sub-process in the data integrity procedure in WPA2 protocol. It is constructed by combining Tk, Data, AAD, and nonce all together and outputs the encrypted data. This encrypted data is concatenated with MAC header, CCM header, and MIC to form the ciphered MPDU

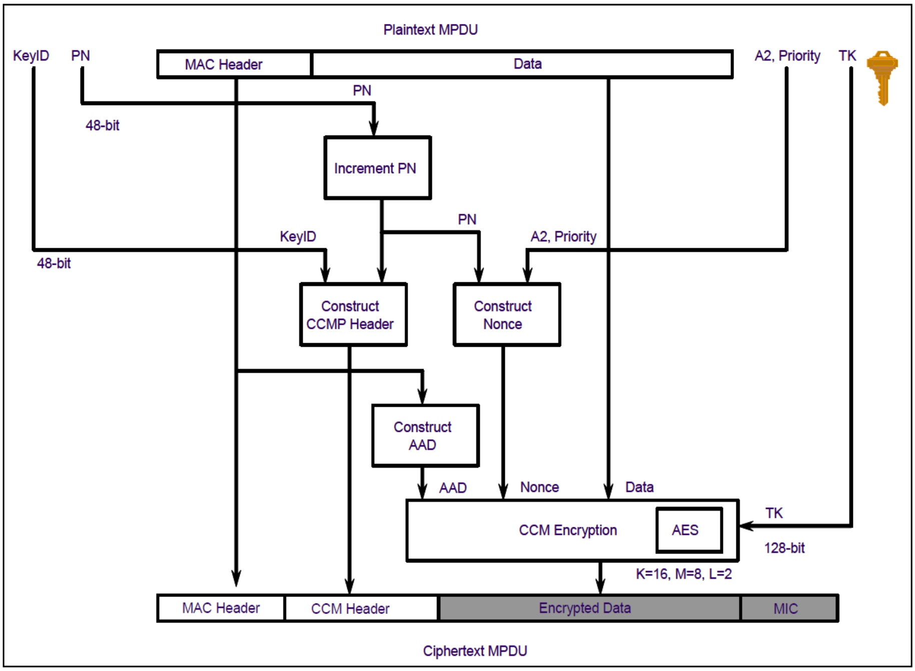


Fig. 5. Integrity in WPA2 [20]

Temporal Key TK changed through each association between the station STA and the access point AP. MIC encrypted with the data is 8 bytes in size. The ciphered frame is sent over the medium and a reversed procedure used to decrypt the encrypted data [5].

4.2 WPA/WPA2 Weaknesses

Although WPA/WPA2 security schemes are strong, there are a number of trivial weaknesses have been found, still, none of them are risky with the security recommendations. However, Authentication mechanisms in WPA-PSK is vulnerable to dictionary attack, which have already been implemented [3]. This attack is based on capturing the 4-way handshake between client and AP which clearly provide enough information to start an attack. Unlike WEP, where statistical methods can be used to speed up the cracking process [3], the Pre-Shared Key (PSK) is derived using the Password Based Key Derivation Function (PBKDF2) which is pseudorandom function that takes several inputs and hashes them multiple times to produce a key [12]. This means that the attacker has all the information and the only thing that the attacker needs is brute force the hand shake to match the 256 bit key which can be a passphrase of 8 to 63 printable ASCII characters. [3].

The Pairwise Transient Key (PTK) is derived from the PMK via the 4-Way Handshake with information used to calculate its value is transmitted in plain text [1]. This information includes MAC address of the client, MAC address of the AP and the two random numbers (ANonce and SNonce) [1]. The only item missing is the PMK/PSK, so the attackers can simply brute force PMK with the need to know the SSID which is easy to be obtained with a sniffer [3].

Even though WPA-PSK is a 256 bits key in length, it has a major weakness because it is based on the pairwise master key (PMK) that is generated by PBKDF2 key derivation function. The PBKDF2 in WPA has five input parameters which are: PMK = PBKDF2 (password, SSID, SSID length, 4096, 256) [1].

Where 4096 is number of the iterations of a sub-function and 256 is length of the output. This means that the strength of PTK relies only on the PMK value, which is the Pre-Shared Key (passphrase) [1].

4.3 WPA/WPA2 Strengths

WPA and WPA2 have several improvements that have helped and supported the Wi-Fi to be more secure than that previously.

As the Clint which is a station (ST) and the access point (AP) have one sharing data cryptography key which is secret between them, the WPA/WPA2 provide mutual authentication in order to prevent the key from being captured when it is transmitted over air [15]. In WPA protocol, data encryption has been improved by using a Temporal Key Integrity Protocol (TKIP). It also has a hashing function that mixes the keys by integrating two components which are the initialization vector (IV) and the base key [17]. Moreover, in order to make sure that the keys have not been changed, the WPA uses an integrity-checking feature. One of the most developments made by WPA and WPA2 are extending the length of Initialization Vector (IV) to 48 bits instead of 24 bits to make sure the IV is not used before, as well as it is used for the TSC (TKIP Sequence Counter) in order to protect against replaying data [15]. In terms of integrity, WPA uses a ‘Michael’, which is a Message Integrity check mechanism (MIC) while WPA2 uses CCM. On the other hand, enterprise mode is another type of WPA/WPA2 modes. Enterprise mode uses 802.1X+EAP for authentication mechanism via an authentication server (802.1x) that offers a perfect control and security to the client’s wireless network traffic. Moreover, in this mode does not use a pre-shard key but it needs a RADIUS server, which is called an authentication server [15]. To avoid reusing the keys there is a rekeying mechanism to afford the freshness of the encrypted plaintext and integrity keys that will be used [10]. One of the most strength things in WPA2 is that it uses an Advanced Encryption Standard (AES) for data encryption. It also uses a block cipher which is working to cipher all blocks of the text every time [8].

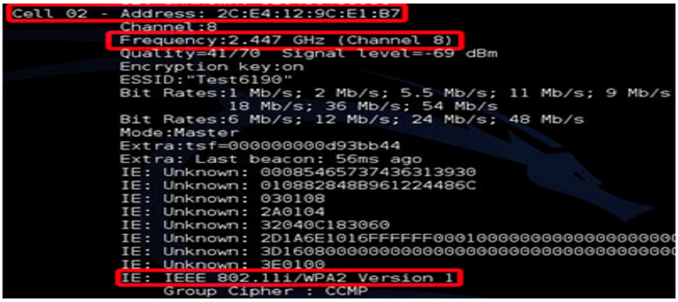
Table 1 summarizes the main differences between WPA and WPA2.

Table 1. Differences between WPA and WPA2

Features of Mechanism	WPA	WPA2
Purpose	Solves the problems that are in WEP protocol	Solves the problems that are in WPA protocol
Require new Hardware	No	Yes
Encryption Cipher Mechanism	RC4 / TKIP	AES/CCMP CCMP/TKIP
Encryption Key Size	128 bits	128 bits
Encryption Key Per Packet	Mixed	No need
Encryption Key Management	802.1x	802.1x
Encryption Key Change	For Each Packet	No need
IV Size	48 bits	48 bits
Authentication	802.1x-EAP	802.1x-EAP
Data Integrity	MIC (Michael)	CCMP
Header Integrity	MIC (Michael)	CCMP
Replay Attack Prevention	IV Sequence	IV Sequence

5 Attack SCENARIO

In the following scenario we will walk through the approach to capture the WPA/WPA2 authentication handshake and then use aircrack-ng to crack the pre-shared key. First, we set up our network target as it appears in Figure 6 which uses WPA2 encryption.

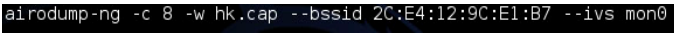


```
Cell 02 - Address: 2C:E4:12:9C:E1:B7
          Channel:8
          Frequency:2.447 GHz (Channel 8)
          Quality=41/70  Signal level=-69 dBm
          Encryption key:on
          ESSID:"Test6190"
          Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 9 Mb/s
                    18 Mb/s; 36 Mb/s; 54 Mb/s
          Bit Rates:6 Mb/s; 12 Mb/s; 24 Mb/s; 48 Mb/s
          Mode:Master
          Extra:tsf=000000000d93bb44
          Extra: Last beacon: 56ms ago
          IE: Unknown: 000854657374363139 30
          IE: Unknown: 010882848B961224486C
          IE: Unknown: 030108
          IE: Unknown: 2A0104
          IE: Unknown: 32040C183060
          IE: Unknown: 2D1A6E1016FFFFFF000100000000000000000000
          IE: Unknown: 3D1608000000000000000000000000000000000000
          IE: Unknown: 3E0100
          IE: IEEE 802.11i/WPA2 Version 1
              Group Cipher : CCMP
```

Fig. 6. The target network

5.1 Four-Way handshake capture

In order to capture the four-way authentication handshake we need to set up our wireless interface to monitor mode. The purpose of this step is to allow the card to monitor the packets received filtering [1].

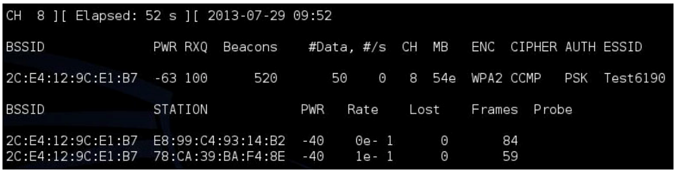


```
airodump-ng -c 8 -w hk.cap --bssid 2C:E4:12:9C:E1:B7 --ivs mon0
```

Fig. 7. Start airodump-ng

Where the parameters are [18]:

- -c 8 is the channel for the wireless network
- --bssid 2C:E4:12:9C:E1:B7 is the access point MAC address. This eliminates extraneous traffic.
- -w hk.cap is the file name prefix for the file which will contain the IVs.
- mon0 is the interface name.
- --ivs is option to save only captured IVs

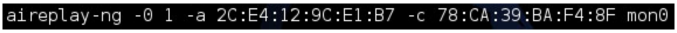


```
CH  8 ][ Elapsed: 52 s ][ 2013-07-29 09:52

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

2C:E4:12:9C:E1:B7  -63 100      520       50    0   8  54e  WPA2 CCMP   PSK  Test6190

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

2C:E4:12:9C:E1:B7  E8:99:C4:93:14:B2  -40    0e- 1      0       84
2C:E4:12:9C:E1:B7  78:CA:39:BA:F4:8E  -40    1e- 1      0       59
```

Fig. 8. Discovering network client

Figure 8 shows what the results look like, there are two wireless clients are connected to the network. To capture the WPA/WPA2 authentication handshake we can perform either active or passive attack [1]. The passive attack simply is to wait for a client to re-associate to the PA. In contrast, the active attack means that a client is forced to de-authenticate in order to accelerate the process [1].

In this case, we are performing active attack using Aireplay-ng which supports various attacks such as deauthentication, to capture WPA handshake data [1].

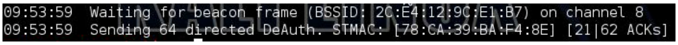


```
aireplay-ng -0 1 -a 2C:E4:12:9C:E1:B7 -c 78:CA:39:BA:F4:8F mon0
```

Fig. 9. Client Deauthentication attack

Where as in [18]:

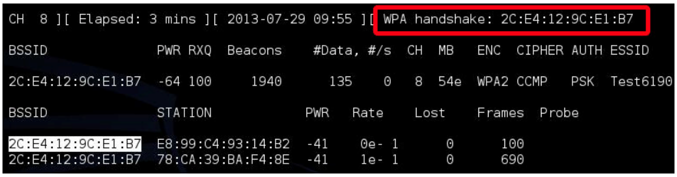
- -0 means de-authentication.
- 1 is the number of de-authenticate to be sent.
- -a 2C:E4:12:9C:E1:B7 is the MAC address of the access point.
- -c 78:CA:39:BA:F48F is the MAC address of the target client that we are de-authenticating.
- mon0 is the interface name.



```
09:53:59  Waiting for beacon frame (BSSID: 2C:E4:12:9C:E1:B7) on channel 8
09:53:59  Sending 64 directed DeAuth. STMAC: [78:CA:39:BA:F4:8E] [21|62 ACKs]
```

Fig. 10. De-authentication output

Figure 11 illustrates what the output looks like of the pervious command. This means that the client is authenticated in order to capture the four-way handshake [1].



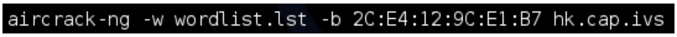
```
CH  8 ][ Elapsed: 3 mins ][ 2013-07-29 09:55 ][ WPA handshake: 2C:E4:12:9C:E1:B7

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

2C:E4:12:9C:E1:B7  -64 100     1940      135    0   8  54e  WPA2 CCMP   PSK  Test6190

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

2C:E4:12:9C:E1:B7  E8:99:C4:93:14:B2  -41    0e- 1      0      100
2C:E4:12:9C:E1:B7  78:CA:39:BA:F4:8E  -41    1e- 1      0      690
```

Fig. 11. Four-Way handshake capture

Also, as a result of the Aireplay-ng command, the four-way handshake is obtained successfully as it shown in figure 11.

5.2 Dictionary Attack

The final step is to launch a dictionary attack, which is a technique for defeating authentication mechanism by trying to each client to determine its passphrase. Aircrack-ng can be used to perform a dictionary attack in order to recover a WPA key [1].



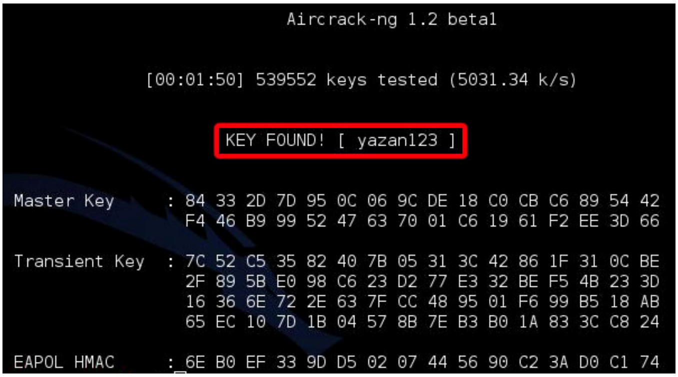
```
aircrack-ng -w wordlist.lst -b 2C:E4:12:9C:E1:B7 hk.cap.ivs
```

Fig. 12. Launching a dictionary attack

In figure 12, the parameters are [18]:

- -w wordlist.lst is the name of the dictionary file.
- --bssid 2C:E4:12:9C:E1:B7 is the access point MAC address
- hk.cap.ivs is name of files that contain the captured four-way handshake.

The purpose of using Aircrack-ng is to duplicate four-way handshake to determine if a particular passphrase in the wordlist matches the results of the four-way handshake [3]. Aircrack-ng takes three inputs ehich are a dictionary file of either ASCII or hexadecimal keys, the mac address of Access point and the file that contains the handshake data [1]. This process is very computationally intensive in practice because it must take each dictionary word and perform 4096 iterations of HMAC-SHA1 before it generates valid PMK. Figure 13 shows that the pre-shared key has been successfully identified [1].



```
                              Aircrack-ng 1.2 beta1


                 [00:01:50] 539552 keys tested (5031.34 k/s)


                          KEY FOUND! [ yazan123 ]


Master Key     : 84 33 2D 7D 95 0C 06 9C DE 18 C0 CB C6 89 54 42
                 F4 46 B9 99 52 47 63 70 01 C6 19 61 F2 EE 3D 66

Transient Key  : 7C 52 C5 35 82 40 7B 05 31 3C 42 86 1F 31 0C BE
                 2F 89 5B E0 98 C6 23 D2 77 E3 32 BE F5 4B 23 3D
                 16 36 6E 72 2E 63 7F CC 48 95 01 F6 99 B5 18 AB
                 65 EC 10 7D 1B 04 57 8B 7E B3 B0 1A 83 3C C8 24

EAPOL HMAC     : 6E B0 EF 33 9D D5 02 07 44 56 90 C2 3A D0 C1 74
```

Fig. 13. Successfully cracking the pre-shared key

Calculating the PMK is very slow since it uses the PBKDF2 algorithm as we mentioned early. The example above has shown that using this technique in Aircrack-ng can check more than 4943 passwords per second. However, it is possible to achieve a time-space tradeoff by pre-computing PMK for given ESSID which is impractical [1].

6 Defence against WPA-PSK Attack

Unfortunately, the vulnerabilities in WPA-PSK authentication, which make the exploit feasible, cannot be avoided [19]. However, there are several steps that can be taken in order to mitigate these vulnerabilities and protect your WLAN against pre-shared keys (PSK) attack which are:

Step 1: avoid using a short PSK that can be guessed too easily or found in a password dictionary. In configuring a passphrase, the IEEE 802.11i standard recommends very strongly to use at least 20 characters [20]. The strongest passphrases which are randomly-generated that mix of lower and uppercase letters, numbers and symbols, the more PSK is protected against dictionary attacks [20].

Step 2: Changing the SSID do not achieve enhanced wireless security but can help to prevent users from accidentally connecting to the wrong WLAN. Also, to make it more difficult for attackers to identify the organization’s WLANs [20].

The dictionary attacks can be infeasible on WPS-PSK by using D-WPA-PSK mechanism which is regular replacement of PSK that are generated by a key generator distributed to all clients in advance [21]. In this method the AP sends a random number to all clients every certain time. The client will send an acknowledgement to the AP when it receives the random number [21]. After all the clients that associated with the AP get random numbers, a new pre-shared keys (PSK) will be generated between the AP and clients which is derived from key generator distributed in advanced [21]. At this point, the clients and the AP will restart the network configuration using the new PSK. This method provides frequent updates of the PSK based on the time that an attacker needs to crack PSK. Therefore d-WPA-PSK achieves somehow enhanced security on a WLAN network [1].

7 Conclusion

The paper described the new protocols developed in Wi-Fi such as WPA and WPA2. Different security services provided by each of them, WPA provides user privacy and confidentiality by using TKIP for encryption and Michael for data integrity. Despite the advantages provided by WPA, it still has some weaknesses regarding the authentication and data integrity processes. Therefore, WPA2 was developed. New mechanism for data integrity in WPA2 was proposed which is CCMP. WPA2 also requires new hardware equipment in order to be installed in. A scenario attack was illustrated in detail and shows that some vulnerability still can take place despite all the security improvements that have been done.

References

1. Mylonas, P., Mavridis, I.P., Androulakis, A.-I.E., Halkias, A.B.: Real-life paradigms of wireless network security attacks (2011)
2. Sukhija, S., Gupta, S.: Wireless Network Security Protocols A Comparative Study (January 2012)
3. Lehembre, G.: Wi-Fi security – WEP, WPA and WPA2 (June 2005)
4. Mathews, M., Hunt, R.: Evolution of Wireless LAN Security Architecture to IEEE 802.11i (WPA2). In: Proceedings of the fourth IASTED Asian Conference on Communication Systems and Networks (2007)
5. Lehembre, G.: —Wi-Fi security –WEP, WPA and WPA2 ||, Article published in number 1/2006 (14) of hakin9 (January 2006), Publication on, `http://www.hsc.fr`
6. Turab, N., Masadeh, S.: —Recommendations guide for WLAN Security. The International Journal of ACM 1(1) (March 2010)
7. Park, S.H., Ganz, A., Ganz, Z.: —Security protocol for IEEE 802.11 wireless local area network. Mobile Networks and Applications 3 (1998)
8. Katz, F.H.: —WPA vs. WPA2: Is WPA2 Really an Improvement on WPA? In: 2010 4th Annual Computer Security Conference (CSC 2010), Coastal Carolina University, Myrtle Beach, SC, April 15-16 (2010)
9. Frankel, S., Eydt, B., Owens, L., Scarfone, K.: Establishing Wireless Robust Security Networks: A Guide to IEEE 802.11i, NIST Special Publication 800-97, National Institute of Standards and Technology (2007)
10. Lashkari, A.H., Danesh, M.M.S., Samadi, B.: FCSIT. A survey on wireless security protocols (WEP, WPA and WPA2/802.11i). In: 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT (2009)
11. Benton, K.: —The evolution of 802.11 wireless security ||, INF 795. UNLV Informatics-Spring (April 18, 2010)

12. Beck, M., Tews, E.: —Practical attacks against WEP and WPA || . In: WiSec 2009: Proceedings of the Second ACM Conference on Wireless Network Security. ACM, New York (2009)
13. Arockiam, L., Vani, B.: —A Survey of Denial of Service Attacks and its Countermeasures on Wireless Network || . International Journal on Computer Science and Engineering 2(5), 1563–1571 (2010)
14. Arockiam, L., Vani, B.: —A Survey of Denial of Service Attacks and its Countermeasures on Wireless NetworkİInternational Journal on Computer Science and Engineering 2(5), 1563–1571 (2010)
15. Bulbul, H.I., Batmaz, I., Ozel, M.: Wireless Network Security: Comparison of WEP (WiredEquivalent Privacy) Mechanism, WPA (Wi-Fi ProtectedAccess) and RSN (Robust Security Network) Security Protocols, e-Forensics 2008, Adelaide, Australia, January 21-23 (2008)
16. Miller, B.: WPA2 Security: Choosing the Right WLAN Authentication Method for Homes and Enterprises. Global Knowledge (2008)
17. Macmichael, J.L.: Auditing Wi-Fi Protected Access (WPA) Pre-Shared Key Mode. Linux Journal (137), 56, 58–60 (2005)
18. Cracking Wireless. Ryan Curtin Ryan at, `http://igglybob.com`
19. De Rango, F., Lentini, D.C., Marano, S.: Static and dynamic 4-wayhandshake solutions to avoid denial of service attack in Wi-Fi protected access and IEEE 802. I li. EURASIP 1. Wire!’ Commun. Netw. (2) (April 2006)
20. Scarfone, K., Dicoi, D., Sexton, M., Tibbs, C.: Recommendations of the National Institute of Standards and Technology. Guide to Securing Legacy IEEE 802.11 Wireless Networks (July 2008)
21. Wang, Y., Jin, Z., Zhao, X.: Practical Defence against WEP and WPA-PSK Attack for WLAN (September 2010)

On the k-error Joint Linear Complexity and Error Multisequence over F_q ($char\ F_q = p$, prime)

M. Sindhu and M. Sethumadhavan*

TIFAC CORE in Cyber Security, Amrita Vishwa Vidyapeetham, Coimbatore, India
{m_sindhu,m_sethu}@cb.amrita.edu

Abstract. Finding fast and efficient algorithms for computing the k-error joint linear complexity and error multisequence of multisequences is of great importance in cryptography, mainly for the security analysis of word based stream ciphers. There is no efficient algorithm for finding the error multisequence of a prime power periodic multisequence. In this paper we propose an efficient algorithm for finding the k-error joint linear complexity together with an error multisequence of m fold prime power periodic multisequences over F_q, where char $F_q = p$, a prime.

Keywords: Word based stream ciphers, Multisequences, Error Joint Linear Complexity, Error multisequence, Generalized Stamp-Martin Algorithm.

1 Introduction

Complexity measures for keystream sequences over finite fields, such as the linear complexity and the k-error linear complexity, is of great relevance to cryptology, in particular, to the area of stream ciphers. Stream ciphers uses deterministically generated pseudorandom sequences to encrypt the message stream. The keystream should be a truly random sequence of elements of a finite field. Security of stream ciphers depends on the quality of the keystreams and the keystream must possess various properties such as having good statistical randomness properties and a high linear complexity in suitable sense, so that the keystream sequence cannot be predicted from a small portion of its terms of the sequence.

The vast majority of proposed keystream generators are based on the use of linear feedback shift registers (LFSR). The length of the shortest LFSR which generates the given sequence is known as the linear complexity of the sequence. A necessary requirement for unpredictability of keystream sequence is long period, which can be attained by large linear complexity. Developments in stream ciphers point towards an interest in word based stream ciphers which require the study of complexity theory of multisequences i.e., of parallel streams of finitely many sequences, and of their complexity properties ([6-8], [10], [12], [14]). A cryptographically strong sequence should not only have a large linear complexity, but also changing a few terms should

[*] Adjunct Scientist, Society for Electronic Transactions and Security (SETS), Chennai, Tamil Nadu, India.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 512–519, 2014.

not cause any significant decrease of the linear complexity. This unfavorable property leads to the concept of k-error linear complexity ([9], [13]). Many authors studied various properties of k-error linear complexity of single and multisequences ([1-5], [7], [10 -14]). In [13] Stamp and Martin gave an efficient algorithm for finding the k-error linear complexity of 2^n periodic binary sequences. This algorithm was later modified by Kaida [2] and he found out the corresponding error vector together with the k-error linear complexity. Kaida et al. in [3] further extended this algorithm to the case of sequences with period p^n over F_q; $Char\ F_q = p$.

There is no efficient algorithm for finding the error multisequence of a prime power periodic multisequence. In this paper we propose an efficient algorithm for the computation of error multisequence e together with the k-error joint linear complexity of m fold multisequences over F_q of period p^n. In other words, we find k-error joint linear complexity and an m fold p^n periodic multisequence $e=(e^{(0)},e^{(1)},...,e^{(m-1)})$ such that

(i) $$\sum_{i=0}^{m-1}\sum_{j=0}^{N-1} e_j^{(i)} \le k, where\ e^{(i)}=(e_j^{(i)}), 0\le j\le N-1, 0\le i\le m-1, 0\le k\le mp^n$$

(ii) The joint linear complexity of the multisequence $S+e$ instead of S is $L_{N,k}(S)$ - the k-error joint linear complexity of S.

Let $S=(S^{(0)},S^{(1)},...,S^{(m-1)})$ with $S^{(h)}=(s_0^{(h)},s_1^{(h)},...,s_{N-1}^{(h)})$, where $s_i^{(h)}\in F_q$, for $0\le h\le m-1$,$0\le i\le N-1$ be an m fold $N=p^n$ periodic multisequence over the finite field $F_q=\{\alpha_0=0,\alpha_1,...,\alpha_{q-1}\}$ of q elements. The joint linear complexity of this multisequence S is defined as the smallest integer $L\ge 0$ for which there exists coefficients $d_1,d_2,...,d_L$ in F_q such that $s_N^{(h)}+d_1 s_{N-1}^{(h)}+...+d_L s_{N-L}^{(h)}=0$ for each $0\le h\le M-1$ and for every $N\ge L$.

An m fold N periodic multisequence S can be interpreted as an $m\times N$ matrix over F_q . For defining the k-error joint linear complexity of multisequences, we need the following definition of term distance [6].

1.1 Definition 1

Let $S=(S^{(0)},S^{(1)},...,S^{(m-1)})$ and $T=(T^{(0)},T^{(1)},...,T^{(m-1)})$ be two m fold N periodic multisequences over F_q .We define the term distance $\delta_T(S,T)$ between S and T as the number of entries in S that are different from the corresponding entries in T.

1.2 Definition 2

Let $S=(S^{(0)},S^{(1)},...,S^{(m-1)})$ be an m fold N periodic multisequences over F_q. For an integer k ($0\le k\le mN$), the k-error joint linear complexity $L_{N,k}(S)$ of S is defined as the smallest possible joint linear complexity obtained by changing k or fewer terms of S in its first period of length N and then continuing the changes periodically with period N. In other words

$$L_{N,k}(S) = \min_T L(T) \tag{1}$$

where the minimum is taken over all m fold N periodic sequence T over F_q with term distance $\delta_T(S,T) \le k$.

2 Algorithm for Computing the *k*-error Joint Linear Complexity and Error Multisequence

Let $S = (S^{(0)}, S^{(1)}, ..., S^{(m-1)})$ be an m fold N periodic multisequences over F_q. Let $N = p^n = pM$ and we write one period of the multisequence S as

$$S_N = (S(0)_M, S(1)_M, ..., S(p-1)_M)$$

where

$$S(j)_M = (S^{(0)}(j)_M, S^{(1)}(j)_M, ..., S^{(m-1)}(j)_M)$$

with

$$S^{(h)}(t)_M = (S^{(h)}_{tM+1}, S^{(h)}_{tM+2}, ..., S^{(h)}_{(t+1)M}), \quad 0 \le t \le p-1, 0 \le h \le m-1.$$

Let $a = (a_{0,} a_1, ..., a_{p-1}) \in F_q^{\,p}$. Define a function F_u on $F_q^{\,p}$ for $0 \le u \le p-1$ as follows [14]

$$F_u(a) = \sum_{t=0}^{p-u-1} \binom{p-t-1}{u} a_t \tag{2}$$

Now define a multisequence $b_{M,u}$, $0 \le u \le p-1$ where each single sequence is of length M as

$$b_{M,u} = (b^{(0)}_{M,u}, b^{(1)}_{M,u}, ..., b^{(m-1)}_{M,u})$$

where

$$b^{(j)}_{M,u} = F_u(S^{(0)}(j)_M, S^{(1)}(j)_M, ..., S^{(p-1)}(j)_M) \tag{3}$$

and F_u is computed component wise.

For the computation of k-error joint linear complexity of S, we are forcing the value of ω as large as possible in the algorithm, so that $(p\text{-}\omega)M$ becomes as small as possible in Step II(4) of our algorithm, under the assumption that the necessary and sufficient condition for minimum number of changes in the original multisequence is less than or equal to k, obtaining the minimal case ω. This criterion can be achieved with introduction of the cost matrix $A_N = (A^{(0)}_N, A^{(1)}_N, ..., A^{(m-1)}_N)$ where $A^{(h)}_N$ is a matrix of size $q \times N$ defined as

$$A_N^{(h)} = \lfloor A^{(h)}(l,j)_N \rfloor, 0 \le l \le q-1, 0 \le h \le m-1, 0 \le j \le N-1 \tag{4}$$

Initially the cost matrix is defined as $A_N^{(h)} = \lfloor A^{(h)}(l,j)_N \rfloor$ with

$$A^{(h)}(l,j)_N = \begin{cases} 0 & if\ \alpha_l = s_j^{(h)} \\ 1 & if\ \alpha_l \ne s_j^{(h)} \end{cases} \tag{5}$$

Proposed algorithm has *n* rounds. In each round, a multisequence $S_M = (S^{M(0)}, S^{M(1)}, ..., S^{M(m-1)})$ where $S^{M(h)} = (s_0^{M(h)}, s_1^{M(h)}, ..., s_{M-1}^{M(h)})$, with each sequence of length *M* is computed from a multisequence S_{pM} with each sequence of length pM where $M = p^{n-r}, 1 \le r \le n$. Also new cost matrices $\left(A_M^{(h)}\right)$ with $A_M^{(h)} = [A^{(h)}(l,j)_M]$ are computed in each step where $A^{(h)}(l,j)_M$ is the minimum number of changes required in the original sequence $S^{(h)}$ of length *N* for changing $s_j^{M(h)}$ to $s_j^{M(h)} + \alpha_l$ in the sequence $S^{M(h)}$ without altering the previous results.

Let $CB_{(M)}^{(h)}$ denote the costs of $b_{M,u}^{(h)}, 0 \le u \le p-2, 0 \le i \le M-1, 0 \le h \le m-1$ and it is given by $CB_{(M)}^{(h)} = \lfloor \lambda^{(h)}(u,i)_M \rfloor$ where $\lambda^{(h)}(u,i)_M$ is the minimum number of changes in $S^{N(h)}$ necessary and sufficient for making

$$b^{(h)}(0)_{M,u} = b^{(h)}(1)_{M,u} = ... = b^{(h)}(u)_{M,u} = 0,\ 0 \le i \le M-1.$$

The total cost of $b_{M,u}^{(h)}$ is defined as $TB_{M,u}^{(h)} = \sum_{i=0}^{M-1} \lambda^{(h)}(u,i)_M$ with

$$\lambda^{(h)}(u,i)_M = \min\left\{ \sum_{j=1}^{p-1} A^{(h)}(e_j^{(h)}, jM+i)_{pM} \middle| e^{(h)} \in D_j^{(h)}(M,u) \right\} \tag{6}$$

where

$$D_j^{(h)}(M,u) = \left\{ e^{(h)} \middle| F_t(e^{(h)} + b_j^{(h)}(M,t)) \right\} = 0, 0 \le t \le u \tag{7}$$

is the set of all $e^{(h)}$ which can make $b^{(h)}(0)_{M,u} = b^{(h)}(1)_{M,u} = ... = b^{(h)}(i)_{M,u} = 0$. Now define

$$TB_{M,u} = \sum_{h=1}^{M} TB_{M,u}^{(h)}, 0 \le u \le p-2 \tag{8}$$

Using $TB_{M,u}$ and the value *k* of allowed term changes, choose the case ω as large as possible to make the increment $(p-\omega)M$ to the *k*-error joint linear complexity of *S* as minimum as possible in Step II(4) . If we can force case ω to happen, then the change vector of S_M is recorded as $(VC_0^{(0)}(M), VC_0^{(1)}(M), ..., VC_0^{(m-1)}(M))$, where

$$VC_0^{(h)}(M) = (V^{(h)}(0,0)_M, V^{(h)}(0,1)_M, \ldots, V^{(h)}(0, pM-1)_M) \tag{9}$$

such that

$$\sum_{j=0}^{p-1} A^{(h)}(V^{(h)}(0, jM+i)_M, jM+i)_{pM} = \begin{cases} \lambda^{(h)}(p-\omega-1)_M, if\ 1 \le \omega \le p-1 \\ \min\left\{ \sum_{j=0}^{p-1} A^{(h)}(e_j^{(h)}, jM+i)_{pM} \middle| e^{(h)} \in F_q^p \right\}, \\ if\ \omega = p \end{cases} \tag{10}$$

Here $V^{(h)}(0, jM+i)_M$ computes the change value of $S^{pM(h)}$ with the minimum number of changes in the sequences $S^{(h)}$ so that the case ω to happen is not getting altered in the algorithm. We also computes the change matrix $V_M^{(h)} = \lfloor V^{(h)}(l,i)_M \rfloor$ such that

$$A^{(h)}(l,i)_M = \sum_{j=0}^{p-1} A^{(h)}(V^{(h)}(l, jM+i)_{pM} \tag{11}$$

During the computation, the values are changed as

$$A^{(h)}(l,i)_M = \min\left\{ \sum_{j=0}^{p-1} A^{(h)}(e_j^{(h)}, jM+i)_{pM} \middle| e^{(h)} \in \hat{D}^{(h)}(l,i)_{M,\omega} \right\} \tag{12}$$

where

$$\hat{D}^{(h)}(l,i)_{M,j} = \left\{ e^{(h)} \middle| \begin{array}{l} F_s(e_0^{(h)}, e_1^{(h)}, \ldots, e_{p-1}^{(h)}) + b^{(h)}(s,i)_M = 0,\ 0 \le s\, p-\omega-1\ and \\ F_{p-\omega}(e_0^{(h)}, e_1^{(h)}, \ldots, e_{p-1}^{(h)}) = F_{p-\omega}(V^{(h)}(0,i)_M, \ldots, V^{(h)}(0,(p-1)M+i)_M) + \alpha_l \end{array} \right\}, \tag{13}$$
$$if\ 1 \le \omega \le p-1$$

and

$$\hat{D}^{(h)}(l,i)_{M,p} = \left\{ e^{(h)} \middle| F_0(e_0^{(h)}, e_1^{(h)}, \ldots, e_{p-1}^{(h)}) = F_0(V^{(h)}(0,i)_M, \ldots, V^{(h)}(0,(p-1)M+i)_M) + \alpha_l \right\}, \tag{14}$$
$$\text{if } \omega = p$$

and

$$S_M^{(h)} = F_{p-\omega}(S^{(h)}(0)_{pM} + VC_0^{(h)}(M), \ldots, S^{(h)}(p-1)_{pM} + VC_{p-1}^{(h)}(M)) \tag{15}$$

with

$$VC_j^{(h)}(M) = (V^{(h)}(0, jM)_M, V^{(h)}(0, jM+1)_M, \ldots, V^{(h)}(0,(j+1)M-1)_M) \tag{16}$$
$$for\ 0 \le j \le p-1$$

For the computation of error multisequence e, we compute the error matrix as

$$E(M) = (E^{(h)}(l,i)_M), 0 \le h \le m-1, 0 \le i \le N-1, 0 \le j \le q-1 \tag{17}$$

where $E^{(h)}(l,i)_M$ is defined as follows. For changing $s_i^{M(h)}$ to $s_i^{M(h)} + \alpha_l$ under the situation that the happening of case ω at M^{th} step is not altered in the algorithm, we can change elements $s^{N(h)}_{jM+i}$ by $E^{(h)}(l, jM+i)$ in the original sequence $S^{(h)}$. But from equation (11) we get

$$E^{(h)}(l, pyM+jM+i)_M = E^{(h)}(V^{(h)}(l, jM+i)_M, pyM+jM+i)_{pM} \quad , \tag{18}$$

$$0 \le l \le q-1, 0 \le j \le p-1, 0 \le h \le m-1, 0 \le y \le \frac{N}{pM}-1, 0 \le i \le M-1$$

For each $0 \le h \le m-1$ the error sequences are initialized as $E^{(h)}(l,i)_N = \alpha_l$. Now we are presenting the algorithm.

Algorithm:

```
I. (a) Initialize $N = p^n$, $S = S_N = (S(0)_N, S(1)_N, ..., S(p-1)_N)$, $M = N$, $L_{N,k} = 0$
   (b) Cost matrix initialization, $A_N^{(h)} = \lfloor A^{(h)}(l, j)_N \rfloor$ using equation (5)
   (c) Error matrix initialization, $E(N) = (E^{(h)}(l,i)_N)$ $with$ $E^{(h)}(l,i)_N = \alpha_l$ using
       equation (17)
II. Round $r$ : for $1 \le r \le n$
      1. $M = \frac{M}{p}$
      2. For $0 \le h \le m-1$ and $1 \le u \le p-2$
           (i)   Compute $b_{M,u}^{(h)}$ from $S(i)_{pM}$ using equation (3)
           (ii)  Compute $TB_{M,u}^{(h)}$ from $S(i)_{pM}$ and $A_M^{(h)}$ using equation (6)
           (iii) Compute $TB_{M,u}$ using equation (8)
       3. Select one of the following $p$ cases
           (i)   Case 1 : $if\ k < TB_{M,0}$
           (ii)  Case $\omega$: if $TB_{M,\omega-2} \le k \le TB_{M,\omega-1}$, $2 \le \omega \le p-1$
           (iii) Case $p$ : if $TB_{M,p-2} \le k$
       4. If Case $\omega$: $L_{N,k} = L_{N,k} + (p-\omega)M$
       5. If Case $\omega$ : $\omega$ = 1 or $p$ ,then $S_M^{(h)} = b_{M,\omega-1}^{(h)}$ and $A_M^{(h)} = A_{M,\omega}^{(h)}$ ,
          $0 \le h \le m-1$ using equation (12)
       6. If Case $\omega$: $2 \le \omega \le p-1$
           For $0 \le h \le m-1$,
                if $TB_{M,\omega-2}^{(h)} < TB_{M,\omega-1}^{(h)}$ then $S_M^{(h)} = b_{M,\omega-1}^{(h)}$ and $A_M^{(h)} = A_{M,\omega}^{(h)}$ ,
                  $0 \le h \le m-1$ using equation (12)
```

else $S_M^{(h)} = 0$, $A_M^{(h)} = 0$ and $k = k - TB_{M,w-2}^{(h)}$

Compute the error matrix $E(M)$ from $E(pM)$ and $V_M^{(h)}$ using equation (18)

III. If $\sum_{h=0}^{m-1} A^{(h)}(l_t,1)_1 \le k$ and $S_1 = (\alpha_{l_0}, \alpha_{l_1}, ..., \alpha_{l_{m-1}})$ for $0 \le l_t \le q-1$, then $L_{N,k}(S) = L_{N,k}$ and $e = (e^{(h)})$ *where* $e^{(h)} = (E^{(h)}(t,0)_1, E^{(h)}(t,1)_1, ...,$ $E^{(h)}(t, N-1)_1)$ where e is an error multisequence

IV. Else if $\sum_{h=0}^{m-1} A^{(h)}(l_t,1)_1 > k$ then $S_1 = (\alpha_{l_0}, \alpha_{l_1}, ..., \alpha_{l_{m-1}})$, $L_{N,k}(S) = L_{N,k} + 1$ and $e = (e^{(h)})$ *where* $e^{(h)} = (E^{(h)}(0,0)_1, E^{(h)}(0,1)_1, ..., E^{(h)}(0, N-1)_1)$ where e is an error multisequence

V. The final $L_{N,k}(S)$ is the *k*-error Joint Linear Complexity of given *m* fold *N* periodic multisequence and e is an error multisequence

From the above discussion we can see that the correctness of this algorithm follows from that of the Generalized Stamp Martin Algorithm [2]. The time complexity of this algorithm is *m* times that of the time complexity of Generalized Stamp Martin Algorithm when applied on a single sequence.

3 Conclusion

There is no efficient algorithm for finding the error multisequence of a prime power periodic multisequence. In this paper we derived an algorithm for finding the *k*-error joint linear complexity and an error multisequence of an *m* fold prime power periodic multisequence over F_q . Finding the error joint linear complexity spectrum and its properties of periodic multisequences over a finite field is also of related interest.

References

1. Ding, C., Xiao, G., Shan, W. (eds.): The Stability Theory of Stream Ciphers. LNCS, vol. 561. Springer, Heidelberg (1991)
2. Kaida, T.: On Algorithms for the *k*-Error Linear Complexity of Sequences over *GF*(p^m) *with Period* p^n, Ph. D. Thesis, Kyusu Institute of Tech. (1999)
3. Kaida, T., Uehara, S., Imamaura, K.: An Algorithm for the k-Error Linear Complexity of Sequences over GF(p^m) with period p^n, p a prime. Information and Computation 151(1-2), 137–147 (1999)
4. Kaida, T.: On the generalized Lauder Paterson algorithm and profiles of the k-error linear complexity for exponent periodic sequences. In: Helleseth, T., Sarwate, D., Song, H.-Y., Yang, K. (eds.) SETA 2004. LNCS, vol. 3486, pp. 166–178. Springer, Heidelberg (2005)

5. Lauder, A.G.B., Paterson, K.G.: Computing the Error Linear Complexity Spectrum of a Binary Sequence of Period 2^n. IEEE Transactions on Information Theory 49(1), 273–281 (2003)
6. Meidl, W.: Discrete Fourier Transform, Joint Linear Complexity and Generalised Joint Linear Complexity of Multisequences. In: Helleseth, T., Sarwate, D., Song, H.-Y., Yang, K. (eds.) SETA 2004. LNCS, vol. 3486, pp. 101–112. Springer, Heidelberg (2005)
7. Meidl, W., Niederreiter, H., Venkateswarlu, A.: Error linear complexity Measures for Multisequences. Journal of Complexity 23(2), 169–192 (2007)
8. Meidl, W., Niederreiter, H.: The expected value of the joint linear complexity of periodic multisequences. Journal of Complexity 19(1), 61–72 (2003)
9. Niederreiter, H.: Linear Complexity and Related Complexity Measures for Sequences. In: Johansson, T., Maitra, S. (eds.) INDOCRYPT 2003. LNCS, vol. 2904, pp. 1–17. Springer, Heidelberg (2003)
10. Sethumadhavan, M., Sindhu, M., Srinivasan, C., Kavitha, C.: An algorithm for k-error joint linear complexity of binary multisequences. Journal of Discrete Mathematical Sciences & Cryptography 11(3), 297–304 (2008)
11. Sethumadhavan, M., Yogha Laxmie, C., Vijaya Govindan, C.: A construction of p-ary balanced sequence with large k-error linear complexity. Journal of Discrete Mathematical Sciences and Cryptography 9(2), 253–261 (2006)
12. Sindhu, M., Sethumadhavan, M.: Linear Complexity Measures of Binary Multisequences. International Journal of Computer Applications 16, 6–10 (2013)
13. Stamp, M., Martin, C.F.: An Algorithm for the k-Error Linear Complexity of Binary Sequences with Period 2^n. IEEE Trans. Inf. Theory 39, 1398–1407 (1993)
14. Venkateswarlu, Studies on error linear complexity measures for multisequences, PhD thesis (2007)

Implementation of an Elliptic Curve Based Message Authentication Code for Constrained Environments

P. Jilna and P.P. Deepthi

National Institute of Technology, Calicut

Abstract. This paper presents the hardware implementation of a new method for message authentication based on elliptic curves. The proposed method makes use of the elliptic curve point multiplication unit already available in the system as a part of key exchange. The point multiplication unit is time shared for generating the authentication code resulting in reduced hardware complexity. Hence it is suitable for applications with limited resources like wireless sensor networks and smart grid. The security of the proposed MAC is vested in Elliptic Curve Discrete Logarithm Problem(ECDLP).

1 Introduction

Verifying the integrity and authenticity of the received data is a prime necessity in communication networks. This is done using Message Authentication Codes. A Message Authentication Code (MAC) is a function which takes as input the message and a secret key that is shared between the communicating parties to return a authentication tag. This tag is appended to the message on transmission and recomputed at the receiver side for authenticating the received data. MAC's have most commonly been constructed out of block ciphers. Another approach is to key the cryptographic hash function [1]. The major obstacle in designing a MAC with cryptographic hash is that hash functions are not keyed primitives which is sharp contrast to MAC function, which uses a secret key as an inherent part of its definition. The security of such MAC's depends on the strength of the underlying hash function. In 2001 NIST published this as a standard known as Keyed - Hash Message Authentication Code (HMAC). The security of the HMAC is increased by truncating the output [2].

MD5 and SHA are the common cryptographic hash functions used for the implementation of keyed hash. As a result the hardware complexity of such MAC's is the same as that of the underlying hash function. The hardware implementation results of the hash functions available in literature shows that it is not suitable for resource constrained environments [3][4].

Present data networks make use of Elliptic Curve Cryptography (ECC) for key exchange. This is because of the increased security per bit of the key. Point multiplication is the cryptographic operation on elliptic curves. Since key exchange is an inevitable part of any data network the EC point multiplication

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 520–529, 2014.

unit will be available in all systems that make use of ECC for key exchange. If a MAC can be generated based on the EC point multiplication unit already available within the system other than implementing an independent module, it will be highly acceptable for applications with limited resources because of the reduced complexity. A MAC with reduced structural complexity based on elliptic curves is proposed and implemented in this paper.

2 Mathematical Background

Elliptic Curves over a field F are set of points (x, y) that satisfy the Weierstrass equation. In general, cubic equations for elliptic curves (Weierstrass equation) take the form

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

The variables x, y and the constants a_1, a_2, a_3, a_4 and a_6 range over any given algebra that meet field axioms. Also included in the definition of an elliptic curve is a single element denoted by O and called the 'point at infinity' or the 'zero point'. Points on an elliptic curve form an abelian group under an addition operation, with O as identity element.

2.1 Elliptic Curves Over $GF(2^m)$

Elliptic curves defined over $GF(2^m)$(Galois Field) have great significance as they are very suitable for hardware implementation. Elliptic curves over $GF(2^m)$ are defined by a cubic equation in which the variables and coefficients take on values in $GF(2^m)$. So, all mathematical operations on EC are performed using the rules of arithmetic in $GF(2^m)$. Since the characteristic of the finite field $GF(2^m)$ is 2, the equation (1) can be transformed by suitable change of variables to get the following forms

$$y^2 + xy = x^3 + a_2x^2 + a_6 \tag{2}$$

$$y^2 + a_3y = x^3 + a_4x + a_6 \tag{3}$$

2.2 Point Multiplication on Elliptic Curves

For a point P on the elliptic curve and an integer 'k', computing a new point 'kP' on elliptic curve is called point multiplication operation on EC. The EC point multiplication is computed by repeated point additions and doubling. The EC operations in turn are composed of basic operations in the underlying finite field (FF or GF). Given k and P, computing R is quite easy. But, at the same time, computing k from R and P is extremely difficult. This is called the discrete logarithm problem for elliptic curves, which is the underlying mathematical problem that provides security and strength to EC-based crypto-schemes [5].

2.3 Truncation Point Problem

The security of an EC based system can be increased by truncating the output i.e, if n is the number of bits in the x-co-ordinate representation of an elliptic curve point then it is truncated to k bits where k $<$ n. This is known as the truncation point problem. TPP states that, given a k bit string, it is hard to tell if it was generated by truncating the x-co-ordinate of a random elliptic curve point or if it was chosen uniformly at random. Security analysis of TPP on elliptic curves defined over various fields is available in literature [6]-[8].

2.4 Message Authentication Codes

Message authentication codes are constructed using hash functions by keying it with a secret key. The basic construction of a cryptographic hash function based MAC is the nested MAC(NMAC). Let G and H be two hash families with key space K and L respectively. Then NMAC of message 'x' is $h_L(g_K(x))$ where $h \in H$ and $g \in G$. A nested MAC is secure if the following two conditions are satisfied.

1. H is a secure MAC, given a fixed unknown key.

2. G is collision resistant, given a fixed unknown key,

3 Proposed Method for Message Authentication

In the proposed method the message M to be authenticated is treated as a polynomial M(x). A modular division operation is performed on M(x) using two polynomials $g_1(x)$ and $g_2(x)$ of degree n to generate the n bit residues r_1 and r_2. The polynomials $g_1(x)$ and $g_2(x)$ are kept secret. The degree n determines the compression ratio. Compute 'r'as $r = r_1 + r_2$. This forms the first phase of MAC generation.Mapping the message in two dimension using two different polynomials decreases the collision probability. In the second phase r is point multiplied with a point P on the elliptic curve and an integer 'k' to generate a new point $R = rkP$. k is an arbitrary integer which is kept secret. A truncation function is applied on the x-co-ordinate of R to generate the MAC.

Security of the use of modular division with secret polynomials for message authentication is well analysed [9]. In addition to this the proposed method increases the security by incorporating a secure one way function of point multiplication. Thus in the proposed method the security lies not only in the secrecy of the polynomials used for modular division but also in ECDLP and truncation point problem.

3.1 Algorithm

Let E be a elliptic curve over $GF(2^n)$. P is a generator point on E. k is an integer which is kept secret. Let $M(x)$ be the message polynomial.

1. perform modular division of $M(x)$ using two generator polynomials $g_1(x)$ and $g_2(x)$ of degree n which are kept secret to obtain residues r_1 and r_2 in n bits.
2. compute $r = r_1 + r_2$
3. compute $R = rkP$.
4. h ← tr (X(R))
5. Return (h)

3.2 MAC Attacks

Big MAC Attack. Since the residues of modular division are equiprobable, by birthday paradox, the computational complexity in attacking the first phase of the MAC generation is $O(2^{(n/2)})$. The security of the second phase is vested in ECDLP and TPP. Therefore the complexity in attacking the second phase is $O(2^{(n-1/2)})$. This shows that the probability of success of a big MAC attack is very less.

Little MAC Attack. The output of the query q_i to the little MAC oracle is the truncated version of the x-co-ordinate of a point on the elliptic curve resulting from the point multiplication q_ikP. Because of the security offered by the truncation point problem and ECDLP, for a successful attack on the little MAC, the attacker need to solve ECDLP and TPP which has a computational complexity of $O(2^{n-1/2})$.

Unknown Key Collision Attack. The residues of modular division are equiprobable i.e. if modular division is done using a polynomial of degree $'n'$ then; the probability of each residue is $1/2^n$. By birthday paradox, the average number of attempts before collision is approximately $2^{(n/2)}$. Thus the computational complexity of finding a collision is of the $O(2^{(n/2)})$.

4 Hardware Implementation

The hardware implementation is based on the elliptic curve defined over $GF(2^n)$. Elliptic curve point multiplication is the most costly operation in terms of hardware complexity in the above proposed method. EC point multiplication involves EC point additions and EC point doublings which are implemented using a number of finite field additions, squaring, multiplication and inversion. Elements of the finite field $GF(2^n)$ can be uniquely expressed using n basis vectors. The most common bases are normal bases and polynomial bases. In the hardware implementation the normal bases representation is used because with this representation the GF squaring reduces to a simple cyclic shifting operation which can be easily implemented. As the major concern is the hardware complexity the algorithm used for GF multiplication is the one cited in [10]. GF addition is obtained by simple bitwise XORing of the two inputs. The most costly hardware operation is the GF inversion which is avoided by representing the points on the elliptic curve in projective co-ordinates. The structure for modular division is implemented using an n bit LFSR. The taps for feedback are determined by the generator polynomial. The algorithm for elliptic curve point multiplication and GF multiplication is given below.

4.1 Algorithm for GF Multiplication

Input $A = (a_0, a_1, a_2, a_3, a_4), B = (b_0, b_1, b_2, b_3, b_4)$. A and B are represented using normal basis $(\beta, \beta^2, \beta^{2^2}, \beta^{2^3}, \beta^{2^4})$ where $\beta \in GF$. Output $S = (s_0, s_1, s_2, s_3, s_4)$.

```
1. Convert A into shifted canonical representation i.e $A = (e_0, e_1, e_2, e_3, e_4)$.
2. Initialize, Sum S= 0
3. for i = 0 to 4
   $B \longleftarrow \beta.B$
   If $e_i = 1$ then S = S + B
   End for
4. Return (S)
```

The hardware structure shown in figure 1 is that of a multiplier defined over $GF(2^5)$ and the primitive polynomial used for the construction is $x^5 + x^4 + x^2 + x + 1$.

The primitive polynomial used for the construction of finite field $GF(2^{16})$ is $x^{16} + x^{15} + x^{13} + x^4 + 1$.

4.2 Algorithm for Elliptic Curve Point Multiplication

Input :An integer k>0, Point P on EC.
Output: Q=k.P.

```
1. Set $k = (k_l \ldots k_1, k_0)_2.(k_l = 1)$
2. Set $Q \longleftarrow P$,
3. for i from l-1 down to 0 do
   $Q \longleftarrow 2Q$,
   if $k_i = 1$
   $Q \longleftarrow P + Q$,
   end if
   end for
4. Return(Q)
```

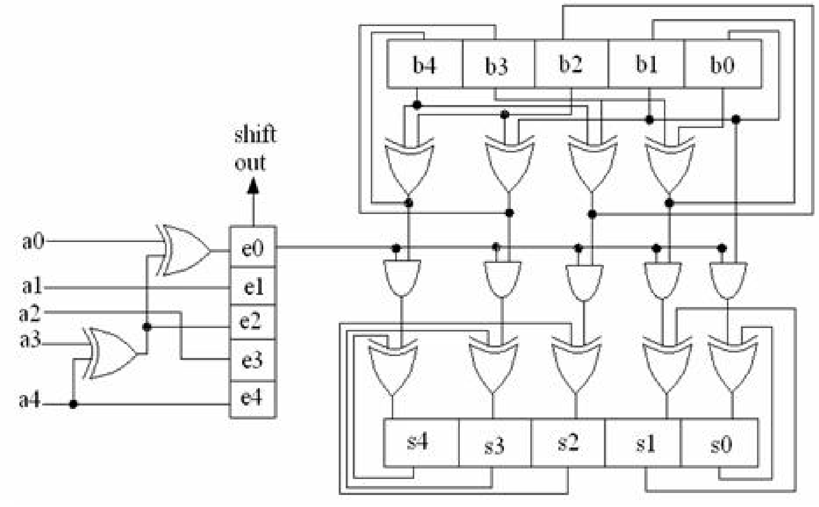


Fig. 1. Hardware structure of GF multiplier

4.3 Elliptic Curve Addition

If $P = (x_1, y_1, z_1)$ and $Q = (x_2, y_2, z_2)$ are the projective equivalents of the points on an elliptic curve, then $P + Q = R = (x_3, y_3, z_3)$ is given by

$$x_3 = AB^2 z_1 z_2 + A^4 \tag{4}$$

$$y_3 = A^2(Bx_1 + Ay_1)z_2 + Cz_1 z_2 + A^3 B \tag{5}$$

$$z_3 = A^3 z_1 z_2 \tag{6}$$

where, $A = x_2 z_1 + x_1 z_2$; $B = y_2 z_1 + y_1 z_2$; $C = A^3 + B^3$.

4.4 Elliptic Curve Doubling

If $P = (x, y, z)$ is the projective equivalent of a point on the elliptic curve then $2P = R = (x_3, y_3, z_3)$ is given by

$$x_3 = AB \tag{7}$$

$$y_3 = x^3 z^3 + Bx^2 + Ayz^3 + z^3 \tag{8}$$

$$z_3 = A^3 \tag{9}$$

where, $A = z^2, B = x^4$.

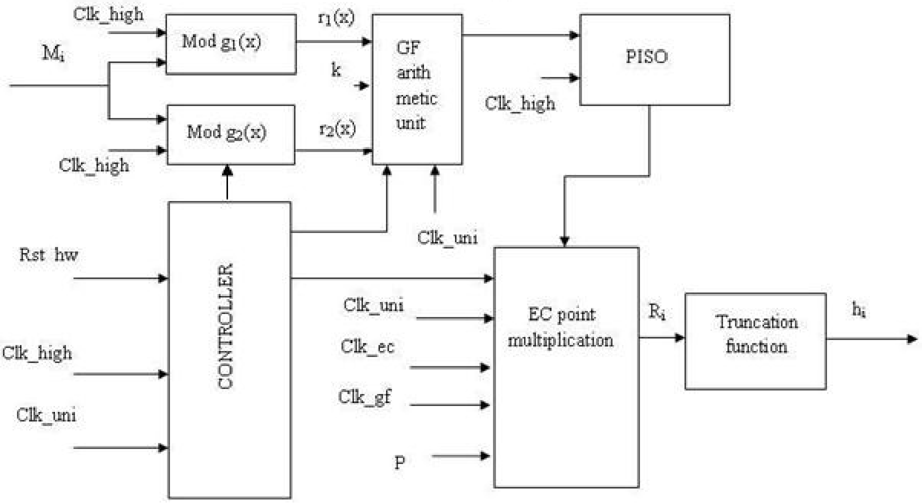


Fig. 2. Structure of the proposed message authentication code

4.5 Implementation

The detailed structure of the proposed message authentication code is shown in figure

The hardware implementation is done using Verilog on Spartan 3A based on elliptic curve defined over $GF(2^{16})$. The hardware requires four different clock frequencies which can be generated either using a clock generation circuit or Digital Clock Manager(DCM) available within the FPGA. A rst-hw pin is provided for hardware reset. The message to be authenticated is given to the two modular division circuits implemented using a 16 bit LFSR. In each clock a message bit is loaded in to the division circuits to generate r_1 and r_2 at the end of the m^{th} clock cycle where m is the number of message bits. Once the modular division is completed the unit is disabled by the control signal from the controller.

The GF arithmetic unit is now enabled and the residues r_1 and r_2 are given to the unit. The GF arithmetic unit consists of a GF multiplier and GF addition circuit. Thus the GF arithmetic unit computes $k(r_1 + r_2)$. The unit requires 16 clock cycles to complete the computation. The result is then loaded to the 16 bit parallel-in serial-out(PISO) register and the GF arithmetic unit is disabled. In each clock the output of PISO is given as input to the point multiplication unit with MSB first.

The point multiplication unit consists of an elliptic curve addition unit and elliptic curve doubling unit. Each of these consists of GF addition, multiplication and squaring units. These units are controlled by an internal controller based on the input bits from the PISO. The various clock inputs to the point multiplication unit are clk-uni, clk-ec, and clk-gf. Clk-gf is the input to GF operation units. Clk-ec controls the data flow between the various GF arithmetic units. Clk-uni controls the data flow between the registers of EC addition and EC doubling units. The resultant of this point multiplication is given to a truncation circuit which outputs the MAC value.

The controller is designed such that each unit is enabled only during their specified time of operation i.e once the modular division of the message string is completed the unit is disabled and then enabled only when the next message string arrives. This helps in reducing the total power consumption of the entire hardware. For a message of size 'm' bits, and a MAC of size 'n' bits, the output MAC is generated in approximately m+2n clock cycles. The RTL view of the MAC generation unit is given below.

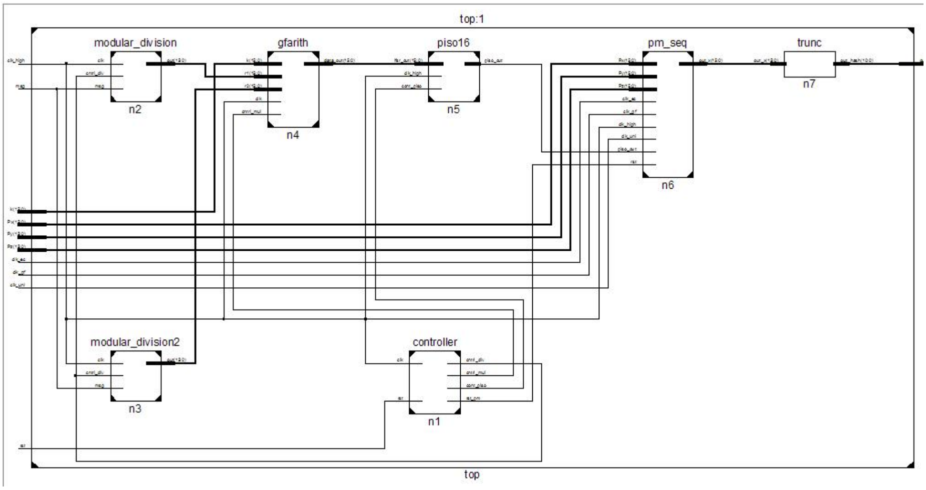


Fig. 3. RTL view of the entire system

4.6 Results

The device utilization summary of the hardware implemented is shown in table 1. From the given table it is clear that the major part of the resources is utilized by the point multiplication unit. The hardware complexity of the system excluding the point multiplication unit is very less. This shows that if point multiplication unit is already available in the system as a part of key exchange then, the same can be time shared with less additional hardware to generate the MAC.

Table 1. Device utilization summary

Logic utilization	Entire unit	EC Point multiplication unit
Number of slice flip flops	1022	902
Number of 4 input LUTs	1307	1131
Number of occupied slices	1074	949

4.7 Comparison with HMAC

HMAC(Hash based MAC) is a derivative of the nested MAC which is standardised by NIST. SHA-1 is the commonly used hashing algorithm for HMAC implementation. SHA-1 is a dedicated algorithm for hashing and the hardware implementation results available in literature shows that structural complexity is high. The proposed method makes use of the point multiplication unit already available in the system as a part of key exchange in a time shared way for MAC generation. From the above table it is clear that the additional hardware requirement is very less when compared to a stand alone algorithm. More over the proposed method is not iterative which results in reduced time complexity in comparison with standard hashing algorithms which are iterative.

5 Conclusion

A new method for message authentication and integrity verification based on elliptic curve point multiplication is proposed. The analysis of various MAC attacks on the proposed method shows that it is computationally secure. The hardware implementation of the proposed system is done using Verilog on Spartan 3A. In comparison with the existing methods, the proposed method is found to have very less structural complexity if time shared with a EC point multiplication unit used for key exchange. This will reduce the overall hardware complexity of the communication system.

Acknowledgement. This work is partially funded by the E-Security division of the Department of Electronics and Information Technology under Ministry of Communication and Information Technology of Government of India.

References

1. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996)
2. National Institute of Standards and Technology, The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication HMAC (2001)

3. Deepakumara, J., Heys, H.M., Venkatesan, R.: Fpga implementation of md5 hash algorithm. In: Canadian Conference on Electrical and Computer Engineering, vol. 2, pp. 919–924. IEEE (2001)
4. Docherty, J., Koelmans, A.: A flexible hardware implementation of sha-1 and sha-2 hash functions. In: International Symposium on Circuits and Systems (ISCAS), pp. 1932–1935. IEEE (2011)
5. Hankerson, D., Menezes, A., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer
6. Chevalier, C., Fouque, P.-A., Pointcheval, D., Zimmer, S.: Optimal randomness extraction from a diffie-hellman element. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 572–589. Springer, Heidelberg (2009)
7. Farashahi, R.R., Pellikaan, R., Sidorenko, A.: Extractors for binary elliptic curves. Designs, Codes and Cryptography 49(1-3), 171–186 (2008)
8. Ciss, A.A., Sow, D.: On randomness extraction in elliptic curves. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011. LNCS, vol. 6737, pp. 290–297. Springer, Heidelberg (2011)
9. Krawczyk, H.: Lfsr-based hashing and authentication. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 129–139. Springer, Heidelberg (1994)
10. Dhanaraj, K.J., Deepthi, P.P., Sathidevi, P.S.: FPGA Implementation of a Pseudo Random Bit Sequence Generator Based on Elliptic Curves. ICGST-Programmable Devices, Circuits and Systems Journal 7(1), 23–31 (2007)

A Hybrid Encryption Technique for Securing Biometric Image Data Based on Feistel Network and RGB Pixel Displacement*

Quist-Aphetsi Kester[1], Laurent Nana[2], Anca Christine Pascu[3],
Sophie Gire[2], Jojo Moses Eghan[4], and Nii Narku Quaynnor[4]

[1] Faculty of Informatics, Ghana Technology University College Accra, Ghana
kquist-aphetsi@gtuc.edu.gh, kquist@ieee.org
[2] Lab-STICC (UMR CNRS 6285), European University of Brittany, UBO, France
[3] HCTI EA 4249 and Lab-STICC (UMR CNRS 6285), European University of Brittany, France
[4] Department of Computer Science, University of Cape Coast, Cape Coast, Ghana

Abstract. Biometric data in a form of images collected from biometric devices and surveillance devices needed to be protected during storage and transmission. Due to the Nature of biometric data, information of the evidence or content of the images need to be preserve after encryption of the plain image and the decryption of the ciphered image. This has to be achieved with a good level of hardness encryption algorithm. Hence this work has proposed a hybrid encryption technique for securing biometric image data based on Feistel Network and RGB pixel displacement. The implementation was done using MATLAB.

Keywords: biometric data, encryption, RGB pixel displacement, image, Feistel Network.

1 Introduction

The collection of biometric data over unsecured and secured channels needed to be protected from third parties. Most surveillance devices collect image data in a form of video or shots of targets from public places for thorough analysis and processing. Lately, the spread of surveillance cameras, high-performance surveillance systems have been attracting much interest in information security and computer vision. Most of these systems have a human tracking and identification capabilities by using soft biometric information [1]. These devices need to secure the evidence they are collecting as well as maintaining the properties of the features of the evidence [2].

This work proposes a new hybrid approach of securing biometric data over unsecured networks by engaging Feistel block ciphering technique and RGB pixel displacement. The paper has the following structure: section 2 covered related

* This work has been supported by LAB-STICC (UMR CNRS 6285) France, AWB-Canada and DCSIT-UCC.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 530–539, 2014.

literature, section 3 covered the Methodology, section 4 explains the algorithms, section 5 is simulated results and analysis, and section 6 concluded the work.

2 Related Works

There has been a lot work over the past years on biometric data in the identification and security of systems. The extensive usage and employment of biometric systems require the concentration on the security holes, by which a reliable system can lose its integrity and acceptance. System access codes like passwords, PIN codes, and biometric inputs suffer from inherent security threats and it is important to pay attention on the security issues before implementing such systems. To solve these problems, Muhammad Khurram Khan and Jiashu Zhang worked on a novel chaotic encryption method to protect and secure biometric templates. They engaged two chaotic maps for the encryption/decryption process. One chaotic map generated a pseudorandom sequence, which was used as private key. While on the other hand, another chaotic map encrypted the biometric data [3].

Image Encryption has gained popularity because of need of secure transmission of images over open networks as there has been an increase in attacks on data flowing over network. Different type of data have different features, that is why for images different techniques have evolved over the years to protect the confidentiality of such data from unauthorized access. Encryption techniques used for encrypting textual data does not work with images, so separate encryption techniques are required for images [4][5].[6][7] developed a cipher algorithm to produce a ciphered image and also to decrypt the ciphered image based on RGB pixel displacement. [8][9][10][11] and [12] employed visual cryptographic technique in encryption of images by encrypting a secret image into some number of shares. Video images transmitted over the internet have seen a tremendous growth over the years [13] and most devices for such transmissions such as remote IP cameras in homes [14], work places, on the street [15], and on phone and smart glasses can easily be used to identify targets and hence need to be highly protected. The increase demand for information gathering and analysis from network devices [16][17] for intelligence purposes [18][19] has triggered the need for advanced and hybrid approaches to securing communication channels and data [20][21][22].

The focus of this work is on the safety and security of biometric images intended for storage or transmission over unsecured networks. Some of these image data are gathered from intelligence public surveillance cameras, crime scene biometric images of suspects etc.

3 Methodology

With the proposed method engaged, Feistel cipher was used to generate a ciphertext. The ciphertext was then used to encrypt the plain biometric images based on the RGB pixel displacement. There was no change of the bit values of the images during the encryption process as well as after it. This gave rise to no pixel expansion in the output image. The image was decomposed based on their RGB pixel values. The R-G-B components were considered as the triplet that formed the characteristics of a pixel of the images used. The ciphering of the image for this research was done by using the RGB pixel values of the images the values were extracted, transposed, reshaped and displaced out of its original pixel positions.

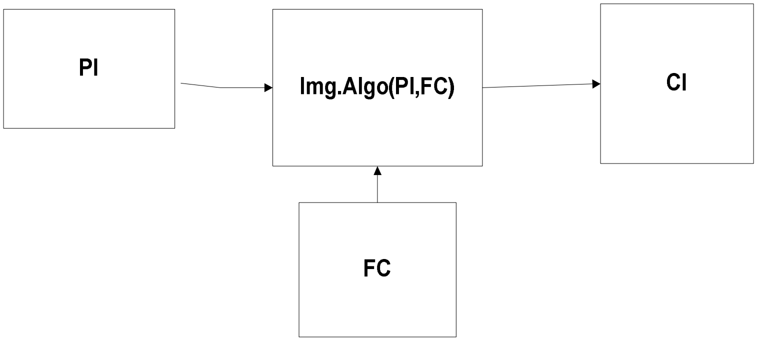


Fig. 1. The Biometric image encryption process

From Fig 1, PI is the plain image, FC is the Feistel cipher, Img.Algo is the image encryption function and CI is the ciphered image.

4 The Algorithms

Feistel ciphering algorithm and the image displacement algorithm is explained below:

4.1 Feistel Cipher

Feistel ciphers are block ciphers, where the ciphertext is calculated by recursively applying a round function to the plaintext.
Let F be the round function and let K_0,K_1,……….. K_n be the sub-keys for the rounds 0, 1…….,n respectively.
Let the basic of operation be as follows:
Split the plaintext block into two equal pieces, (L_0, R_0)
For each round i =0,1,…..,n, compute

$$L_i+1 = R_i. \tag{1}$$

$$R_{i+1}= L_i \oplus F(R_i, K_i). \tag{2}$$

Then the ciphertext is (R_{n+1}, L_{n+1}).

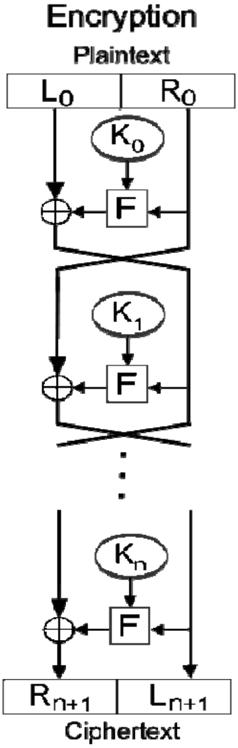


Fig. 2. The Feistel encryption process

4.2 The Image Encryption Process

Step1 Start
Step2. Extraction of data from a plain image,
Let I= an image=f (R, G, B)
I is a color image of m x n x 3 arrays

$$\left(\begin{array}{ccc} R & G & B \\ r_{11} & g_{12} & b_{13} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ r_{m1} & g_{m2} & b_{m3} \end{array}\right) \tag{3}$$

(R, G, B) = m x n
Where R, G, B ∈ I
(R o G) ij = (R) ij. (G) ij

Where r_11 = first value of R

r= [ri1] (i=1, 2… m)

x ∈ r_i1 : [a, b]= {x ∈ I: a ≤ x ≥ b}

a=0 and b=255

R= r= I (m, n, 1)

Where g_12 = first value of G

g= [gi2] (i=1, 2... m)

x ∈ g : [a, b]= {x ∈ I: a ≤ x ≥ b}

a=0 and b=255

G= g= I (m, n, 1)

And b_13 = first value of B

g= [bi3] (i=1, 2... m)

x ∈ b_i1 : [a, b]= {x ∈ I: a ≤ x ≥ b}

a=0 and b=255

B=b= I (m, n, 1

Such that R= r= I (m, n, 1)

Step3. Extraction of the red component as ‘r’

Let size of R be m x n [row, column] = size (R) = R (m x n)

$$\text{rij= r= I (m, n, 1)} = \begin{pmatrix} R \\ r_{11} \\ \vdots \\ \vdots \\ r_{n1} \end{pmatrix} \quad (4)$$

Step4. Extraction of the green component as ‘g’

Let size of G be m x n [row, column] = size (G)

$$\text{gij= g= I (m, n, 1)} = \begin{pmatrix} \mathrm{G} \\ g_{12} \\ \vdots \\ \vdots \\ g_{n2} \end{pmatrix} \quad (5)$$

Step5. Extraction of the blue component as ‘b’

Let size of B be m x n [row, column] = size (B) = B (m x n)

$$\text{bij= b= I (m, n, 1) =} \begin{pmatrix} B \\ b_{13} \\ \vdots \\ \vdots \\ b_{n3} \end{pmatrix} \quad (6)$$

Step6. Getting the size of r as [c, p]

Let size of R be [row, column] = size (r) =r (c x p)

Step7. Engagement of ciphertext (R_{n+1}, L_{n+1}), to iterate the step 8 to step 14 of the image encryption process.

Step8. Let r =Transpose of rij

$$r = \begin{pmatrix} R & & & & \\ r_{11} & \cdots & \cdots & \cdots & r_{n1} \end{pmatrix} \tag{7}$$

Step9. Let g =Transpose of gij

$$g = \begin{pmatrix} G & & & & \\ g_{12} & \cdots & \cdots & \cdots & g_{n2} \end{pmatrix} \tag{8}$$

Step10. Let b =Transpose of bij

$$b = \begin{pmatrix} B & & & & \\ b_{13} & \cdots & \cdots & \cdots & b_{n3} \end{pmatrix} \tag{9}$$

Step11. Reshaping of r into (r, c, p)

$$r = \text{reshape}\ (r, c, p) = \begin{pmatrix} R \\ r_{11} \\ \vdots \\ \vdots \\ r_{1n} \end{pmatrix} \tag{10}$$

Step12. Reshaping of g into (g, c, p)

$$g = \text{reshape}\ (g, c, p) = \begin{pmatrix} G \\ g_{12} \\ \vdots \\ \vdots \\ g_{n2} \end{pmatrix} \tag{11}$$

Step13. Reshaping of b into (b, c, p)

$$b = \text{reshape}(b, c, p) = \begin{pmatrix} B \\ b_{13} \\ \vdots \\ \vdots \\ \vdots \\ b_{n3} \end{pmatrix} \quad (12)$$

Step14. Concatenation of the arrays r, g, b into the same dimension of ‘r’ or ‘g’ or ‘b’ of the original image

$$= \begin{pmatrix} R & G & B \\ r_{11} & g_{12} & b_{13} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ r_{n1} & g_{n2} & b_{n3} \end{pmatrix} \quad (13)$$

Step15. Finally the data will be converted into an image format to get the encrypted image. The inverse of the algorithm will decrypt the encrypted image

5 Results and Analysis

The following image was encrypted based on the following parameters:

The plaintext for the Feistel ciphers = “EncryptionEncryptionEncryption Encryption”

The key for the Feistel ciphers = “positivity”

The Ciphertext for the Feistel ciphers = “eJby45VgtuR2N+vIm2dQHqP5i9Ns87 AcZxir3ikD5MoU2GnnW/CHaEnGiKoHDxUg”

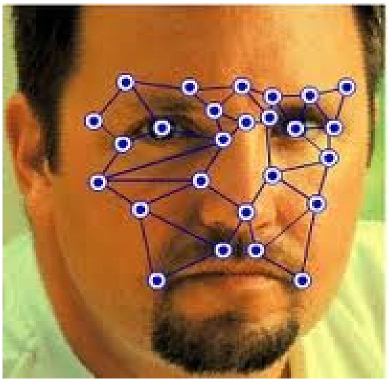


Fig. 3. Plain Image

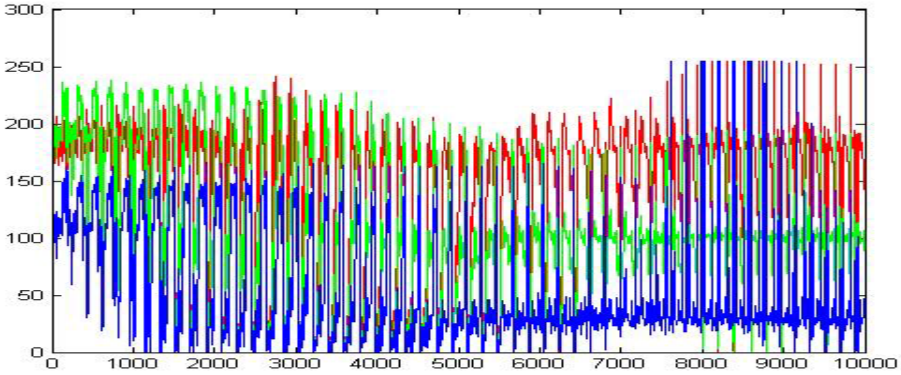


Fig. 4. An RGB graph of the plain Image in fig 3

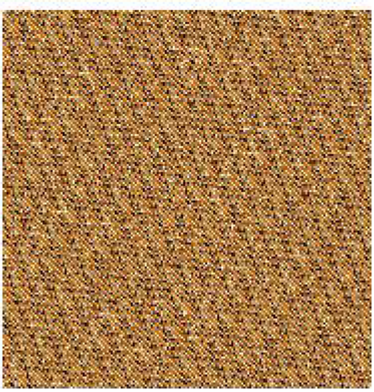


Fig. 5. Ciphered Image of plain Image in fig 3

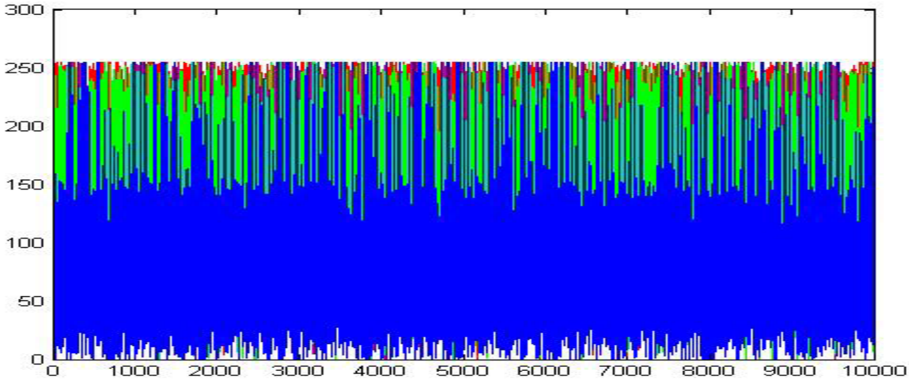


Fig. 6. An RGB graph of the Ciphered Image in fig 5

6 Conclusion

The first 10000 pixel values of the plain image and the ciphered image were plotted as shown in fig 4 and fig 9 respectively. The Entropy value and the arithmetic mean value of the pixels of the plain image and the ciphered image were found to be 7.4937and 118.0559 respectively, this means that there was no pixel expansion after the encryption process. Hence the quality of the image remained the same after decryption.

References

1. Koga, Y., Yamazaki, Y., Ichino, M.: A study on the surveillance system using soft biometric information. In: 2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE), October 1-4, pp. 262–266 (2013), doi:10.1109/GCCE.2013.6664819
2. Hae-M. Moon, C., Won, P.S.B.: The Multi-Modal Human Identification Based on Smartcard in Video Surveillance System. In: 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom), Green Computing and Communications (GreenCom), December 18-20, pp. 691–698 (2010), doi:10.1109/GreenCom-CPSCom.2010.74
3. Khan, M.K., Zhang, J.: Enhancing the transmission security of content-based hidden biometric data. In: Proceedings of the Third international conference on Advances in Neural Networks, Chengdu, China, May 28-June 01 (2006)
4. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)
5. Goel, A., Chandra, N.: A Technique for Image Encryption Based on Explosive n*n Block Displacement Followed by Inter-pixel Displacement of RGB Attribute of a Pixel. In: 2012 International Conference on Communication Systems and Network Technologies (CSNT), May 11-13, pp. 884–888 (2012), doi:10.1109/CSNT.2012.190
6. Kester, Q., Koumadi, K.M.: Cryptographie technique for image encryption based on the RGB pixel displacement. In: 2012 IEEE 4th International Conference on Adaptive Science & Technology (ICAST), October 25-27, pp. 74–77 (2012), doi:10.1109/ICASTech.2012.6381069
7. Zhu, G., Wang, W., Zhang, X., Wang, M.: Digital image encryption algorithm based on pixels. In: 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), vol. 1., October 29-31, pp. 769–772 (2010), doi:10.1109/ICICISYS.2010.5658790
8. Zhi, Z., Arce, G.R., Di Crescenzo, G.: Halftone visual cryptography. IEEE Transactions on Image Processing 15(8), 2441–2453 (2006), doi:10.1109/TIP.2006.875249
9. Liu, F., Wu, C.-K., Lin, X.-J.: Colour visual cryptography schemes. Information Security, IET 2(4), 151–165 (2008), doi:10.1049/iet-ifs:20080066
10. Kang, I., Arce, G.R., Lee, H.-K.: Color Extended Visual Cryptography Using Error Diffusion. IEEE Transactions on Image Processing 20(1), 132–145 (2011), doi:10.1109/TIP.2010.2056376
11. Liu, F., Guo, T., Wu, C., Qian, L.: Improving the visual quality of size invariant visual cryptography scheme. J. Vis. Comun. Image Represent. 23(2), 331–342 (2012), doi:10.1016/j.jvcir.2011.11.003
12. Tzeng, W.-G., Hu, C.-M.: A New Approach for Visual Cryptography. Des. Codes Cryptography 27(3), 207–227 (2002), doi:10.1023/A:1019939020426

13. Gao, W., Tian, Y., Huang, T., Yang, Q.: Vlogging: A survey of videoblogging technology on the web. ACM Comput. Surv. 42(4), Article 15, 57 pages (2010), doi:10.1145/1749603.1749606
14. Hui, S.C., Wang, F.: Remote Video Monitoring Over the WWW. Multimedia Tools Appl. 21(2), 73–195 (2003), doi:10.1023/A:1025520709481
15. Esteve, M., Palau, C.E., Martnez-Nohales, J., Molina, B.: A video streaming application for urban traffic management. J. Netw. Comput. Appl. 30(2), 479–498 (2007), doi:10.1016/j.jnca.2006.06.001
16. Turnbull, B., Slay, J.: Wireless Forensic Analysis Tools for Use in the Electronic Evidence Collection Process. In: 40th Annual Hawaii International Conference on System Sciences, HICSS 2007, p. 267a (January 2007), doi:10.1109/HICSS.2007.617
17. Aisha Al-Abdallah, A., Asma Al-Emadi, A., Mona Al-Ansari, M., Nassma Mohandes, N., Malluhi, Q.: Real-time traffic surveillance using ZigBee. In: 2010 International Conference on Computer Design and Applications (ICCDA), June 25-27, vol. 1, pp. V1-550–V1-554 (2010), doi:10.1109/ICCDA.2010.5540694
18. Zamin, N.: Information Extraction for Counter-Terrorism: A Survey. In: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, COMPUTATIONWORLD 2009, November 15-20, pp. 520–526 (2009), doi:10.1109/ComputationWorld.2009.105
19. Park, H., Ham, Y.-H., Park, S.-J., Woo, J.-M., Lee, J.-B.: Large Data Transport for Real-Time Services in Sensor Networks. In: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, COMPUTATIONWORLD 2009, November 15-20, pp. 404–408 (2009), doi:10.1109/ComputationWorld.2009.97
20. Papalilo, E., Freisleben, B.: Combining incomparable public session keys and certificateless public key cryptography for securing the communication between grid participants. In: Meersman, R. (ed.) OTM 2007, Part II. LNCS, vol. 4804, pp. 1264–1279. Springer, Heidelberg (2007)
21. Bansal, R., Sehgal, P., Bedi, P.: Securing fingerprint images using a hybrid technique. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI 2012), pp. 557–565. ACM, New York (2012), doi:10.1145/2345396.2345488
22. Hasan, R., Sion, R., Winslett, M.: Preventing history forgery with secure provenance. Trans. Storage 5(4), Article 12, 43 pages (2009), doi:10.1145/1629080.1629082

A Simple and Efficient Key-Dependent S-Box Design Using Fisher-Yates Shuffle Technique

Musheer Ahmad[1], Parvez Mahmood Khan[2], and Mohd Zeeshan Ansari[1]

[1] Department of Computer Engineering, Faculty of Engineering and Technology,
Jamia Millia Islamia, New Delhi 110025, India
[2] Department of Computer Science & Engineering, Faculty of Engineering,
Integral University, Lucknow 226026, India

Abstract. The substitution boxes are the only components in most of symmetric encryption systems that induce nonlinearity to provide efficacious data confusion. The cryptographic potency of these systems primarily depends upon the caliber of their S-boxes. This brings new challenges to design cryptographically efficient S-boxes to develop strong encryption systems. Here, a simple and effective method to design an efficient 8×8 S-box is put forward. The proposed design methodology is based on the classical Fisher-Yates shuffle technique. A piece-wise linear chaotic map is incorporated to act as a source to generate random numbers for proficient execution of shuffle technique. The construction of dynamic S-box is under the control of secret key. The performance evaluation of proposed S-box against standard statistical tests like bijective property, nonlinearity, strict avalanche criteria and equiprobable I/O XOR distribution reveals its excellent performance. Moreover, the proposed S-box is also compared with some recent chaos-based S-boxes. The investigations confirm that the design is consistent and suitable for secure communication.

Keywords: Fisher-Yates shuffle, substitution-box, chaotic map, nonlinearity, secure communication.

1 Introduction

Most of the traditional symmetric DES-like encryption systems practically rely on the usage of substitution boxes. They are meant to obscure the relationship between the key and ciphertext data as complex as possible to achieve Shannon's property of confusion. Proficient confusion frustrates the attacker who utilizes the ciphertext statistics to recover the key or the plaintext [1, 2]. The cryptographic potency of these encryption systems primarily depends upon the efficiency of their substitution boxes. An efficient S-box with sound cryptographic features is significant for development of strong cryptographic system. As a result, they constitute the core nonlinear components of the systems. They are the only components capable of inducing the nonlinearity in the security system. A high nonlinearity is desirable since it decreases the correlation between output and the input or a linear combination of the two [3]. An *m*×*n* substitution box is a nonlinear mapping function S: $\{0, 1\}^{m} \rightarrow \{0, 1\}^{n}$, *m* and

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 540–550, 2014.

n need not be equal, which can be represented as $S(x) = [f_{n-1}(x)f_{n-2}(x) \ldots f_1(x)f_0(x)]$, where f_i $(0 \leq i \leq n\text{-}1)$ is a Boolean function defined as f_i: $\{0, 1\}^m \rightarrow \{0, 1\}$. An S-box is keyed or keyless and static or dynamic. The design methodologies and criteria utilized to synthesize S-boxes are equally significant and prevailing. It should have the characteristics of simplicity, low algorithmic complexity and low computational time. The various design methodologies suggested in the literature to construct S-boxes include algebraic techniques [4], heuristic methods [5], power mapping technique [6], cellular automata [7], etc. Substitution boxes based on algebraic techniques are much popular because they have strong cryptographic characteristics and resilient to linear and differential cryptanalysis [4, 8]. However, the recent advances in algebraic cryptanalysis reveal that they are weak against some attacks [9, 10]. Hence, rather than focusing on the design of conventional algebraic techniques based S-boxes, there is a trend of constructing S-boxes based on some other alternatives. A cryptographic efficient S-box should have the characteristics of balancedness, high nonlinearity scores, low maximum differential approximation probabilities, an avalanche effect close to ideal value etc.

Chaotic systems are the dynamical systems that have certain cryptographically desirable features such as high sensitivity to their initial conditions, long periodicity, unpredictability and random-behaviour. The researchers have studied the analogy between the chaotic and cryptographic properties. Their features have attracted the attentions of researchers worldwide. They are extensively exploited to design strong chaos-based security systems to protect multimedia data like images, audio, videos, etc. Unlike traditional encryption systems, the chaos-based systems are considered competent and credential for providing high security with low computational overheads for a real-time secure communication. Nowadays, the features of chaotic systems are also explored to synthesize the nonlinear components i.e. the S-boxes of block ciphers. The researchers are attempting to construct chaos-based S-boxes with strong cryptographic characteristics. Several methods have been suggested in the literature to construct chaos-based S-boxes [11-17]. In this work, a simple and effective method is presented to construct chaotic S-box based on the classical Fisher-Yates shuffle technique. The performance assessment of proposed method demonstrates that the S-box has strong and better cryptographic characteristics.

The organization of rest of the paper is as follows: Section 2 gives the basic description of Fisher-Yates shuffle, chaotic system used and proposed method of designing S-box. The performance of the proposed S-box is quantified and analyzed in Section 3, while the conclusion of the work is made in Section 4.

2 Designing S-Box

The basic design concepts used to construct the proposed S-box are described in the following subsections.

2.1 Fisher-Yates Shuffle

The Fisher–Yates shuffle, suggested by Ronald Fisher and Frank Yates, is an algorithm for generating a random permutation of a finite linear array [18]. The Fisher–Yates shuffle is unbiased, so that every permutation of the array is equally likely. The modern version of the Fisher–Yates shuffle was introduced by Richard Durstenfeld [19] and popularized by Donald E. Knuth in his pioneer book *The Art of Computer Programming* [20]. The modern version is an in-place shuffle, efficient requiring only time proportional to the number of elements being shuffled and no additional storage space.

```
To shuffle an array $S$ of $n$ elements (indices 1...n)
  for $i \leftarrow n$ to 1 do
       $j \leftarrow$ random integer with $1 \le j \le i$
       exchange($S[j]$, $S[i]$)
  end
```

The shuffling effect of the algorithm is solely depends on the quality of random generation of indices *j*. This modern Fisher-Yates shuffle algorithm is employed to shuffle randomly the pre-initialized array of 8×8 S-box elements. The shuffling effect is improved by: (1) incorporating a piece-wise linear chaotic map as source to generate random indices and (2) applying the iterations of algorithm on linear array shuffled so far. Like other chaos-based methods suggested in literature to construct substitution boxes, the proposed method is also simple, has low algorithmic complexity and incurs low computational overheads. However, the proposed methodology synthesizes efficient substitution box with better cryptographic characteristics. To the best of our knowledge and study, no one has yet utilized the Fisher-Yates shuffle algorithm for the construction of cryptographic substitution boxes.

2.2 Piece-Wise Linear Chaotic Map

The piece-wise linear chaotic map is one-dimensional dynamical system that has been considered among the most studied chaotic systems, its system equation is described as follows [21]:

$$x(n+1) = \begin{cases} \dfrac{x(n)}{p} & 0 < x(n) \le p \\ \dfrac{1-x(n)}{1-p} & p < x(n) < 1 \end{cases} \tag{1}$$

Where x is state of the system, n is the number of iterations and $x(n) \in (0,1)$ for all n. The lyapunov exponent of a dynamical system specifies the rate of separation of minutely close trajectories [22]. For discrete dynamical systems $x(n+1) = g(x(n))$, with initial value $x(0)$, it is defined as:

$$\lambda = \lim_{n \to \infty} \frac{1}{n} \sum_{k=0}^{n-1} In\left|g'(x(k))\right|$$

The number of lyapunov exponents is determined by the dimension of the system. If atleast one lyapunov exponents of a system is positive, then the system is considered chaotic. It is evident from Fig. 1(a) that system described in (1) exhibits chaotic behaviour, as it has positive lyapunov exponents for values of control parameter $p\in (0,1)$. The largest positive lyapunov exponent is reckoned at $p = 0.5$. The map's bifurcation diagram depicted in Fig. 1(b) reveals that, for every value of parameter p, the system trajectory of PWLCM $x(n)$ visits the entire interval (0, 1) [23]. It has been studied that PWLCM exhibits better features than logistic map [24]. Unlike logistic map, it has no blank windows (non-chaotic regions) [25]. The initial values assigned to $x(0)$ and p, for execution of system, act as key to control its chaotic behaviour and trajectory. A slight difference in key causes a high deviation in its trajectory. The system is used to generate random real numbers by sampling its system trajectory. Eventually, the key drives the dynamic generation of S-boxes. A number of different efficient S-boxes can be generated by making a minor change in the key. The key-dependency of S-boxes is illustrated in Section 3.

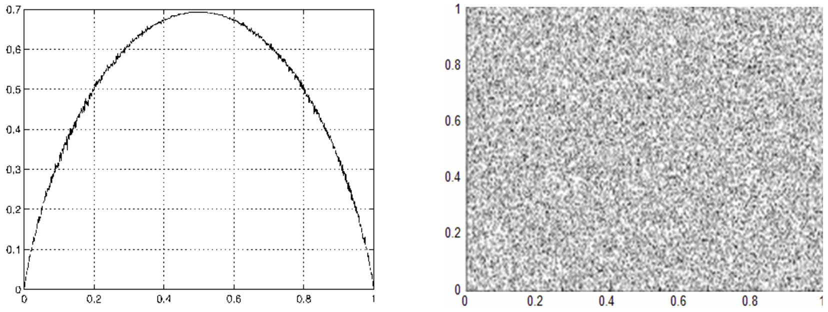


Fig. 1. Piece-wise linear chaotic map plots depicting (a) lyapunov exponent vs p and (b) bifurcation diagram: $x(n)$ vs p

2.3 Proposed Method

The procedure of the proposed S-box design methodology is as follows:

A.1. Initialize a linear array S of size 256 with values starting from 0 to 255 in ascending order. Set $\Delta = length(S)$.

A.2. Iterate the piece-wise linear chaotic map for N_0 times to die-out the transient effect of map with selected initial conditions.

A.3. Set $cnt = 1$.

A.4. Further iterate the map (1) and sample the chaotic state-variable x.

A.5. Extract a random number $m \in [1, k]$ from x as:

$$m = \{floor(x*10^{10})\}mod(k) + 1 \qquad \text{where, } k = \Delta - cnt + 1$$

A.6. Exchange the two elements of array S at positions m and k i.e. $S(k) \leftrightarrow S(m)$.

A.7. Set $cnt = cnt + 1$, If $cnt < 256$ go to step **A.4**.

A.8. Re-apply the Fisher-Yates shuffle on current array S by repeating steps **A.3** to **A.7** for ξ times.

A.9. Translate resultant shuffled linear array S to 16×16 table to get final S-box.

3 Analysis of Proposed S-Box

In order to assess and quantify the cryptographic characteristics of substitution boxes, various statistical measures are developed in the literature [4, 11, 26-28]. It is widely accepted among cryptologists that the significant performance measures are bijectivity, nonlinearity, strict avalanche criteria and equiprobable I/O XOR distributions. Unfortunately, it is not possible to fulfill all of these criteria to the maximum. For example, it is impossible to reach both the balancedness and the highest nonlinearity. The bent-Boolean functions, of size n-bit, can provide the highest possible nonlinearity score of $2^{n-1} - 2^{(n/2)-1}$, but they are not balanced [27]. Thus, some tradeoffs have to be made while designing efficient S-boxes. These statistical parameters are opted and evaluated to judge the performance of proposed S-box to find its suitableness for block encryption. The performance is also compared with few of the recent chaos-based S-boxes. The initial values taken for simulation are: $x(0)$=0.3571, p=0.587, N_o=1000 and ξ=3. The chaotic S-box constructed using proposed method is depicted in Table 1. Moreover, the different S-boxes can be synthesized by slightly modifying the any of the key parameters $x(0)$, p, N_0 and ξ as well. To demonstrate the key-dependency, the proposed method is applied to construct 1000 different S-boxes by updating the initial value of $x(0)$ with an increment of 0.000223 for each S-box every time and analyzed.

3.1 Bijectivity

An n-bit Boolean function f is said to be balanced, if it has the equal number of 0's and 1's. Balancedness is one of significant cryptographic characteristics in the sense that the output of the Boolean function should not leak any statistical information about structure [3]. A Boolean function f_i is bijective if it holds the relation [11]: $wt(\sum_{i=1}^{n} a_i f_i) = 2^{n-1}$, where $a_i \in \{0, 1\}$, $(a_1, a_2, \ldots, a_n) \neq (0, 0, \ldots, 0)$ and $wt(.)$ is hamming weight. It is experimentally verified that all 1000 S-boxes obtained with proposed method satisfy the bijective property.

3.2 Nonlinearity

The nonlinearity of a Boolean function *f* is its minimum distance to all affine functions. A function with low nonlinearity is prone to linear approximation attack. A strong cryptographic S-box should have high measures of nonlinearities. The nonlinearity NL_f of a Boolean function *f(x)* is determined as:

$$NL_f = 2^{n-1}(1-2^{-n}\max| S_{\langle f\rangle}(w)|)$$

$$S_{\langle f\rangle}(w) = \sum_{w\in GF(2^n)}(-1)^{f(x)\oplus x.w}$$

$S_{(f)}(w)$ is the Walsh spectrum of *f(x)* and *x.w* denotes the dot-product of *x* and *w*. Nonlinearity scores for the eight Boolean functions of the proposed S-box are 106, 108, 108, 106, 104, 106, 104, 106 whose *mean* value is 106. These nonlinearity scores are compared with that of existing chaos-based and other S-boxes in Table 2. The proposed S-box provides higher *min* and *mean* value of nonlinearity scores. The *max* value is higher than Wang's, Ozkaynak's, Xyi's, Residue prime's and comparable to other's S-boxes. As can be seen in Fig. 2 that the average nonlinearity of each of 1000 S-boxes is greater than 100 meaning that the proposed S-boxes have high nonlinearity scores.

3.3 Strict Avalanche Criteria

If a Boolean function satisfies the strict avalanche criteria, it means that each output bit should change with a probability of ½ whenever a single input bit is changed. Webster and Tavares developed a procedure to check whether an S-box satisfies the SAC [28]. Following the procedure, a dependency matrix is calculated to test the SAC of the S-box and provided in Table 3. The SAC of the proposed S-box comes out as

Table 1. Proposed Substitution-Box

153	180	218	160	120	182	216	103	93	11	30	237	82	74	106	193
241	56	17	155	116	26	65	32	225	130	69	14	223	99	70	121
0	126	151	19	25	255	207	254	71	21	111	192	219	61	46	145
75	122	31	154	41	200	50	57	142	177	188	235	170	118	58	162
10	91	181	101	55	34	179	249	76	206	83	13	27	148	159	68
150	85	224	199	39	44	12	246	166	98	229	114	94	194	78	96
231	147	209	35	139	48	86	233	36	6	5	33	73	202	123	135
214	227	168	40	201	244	1	234	144	191	208	242	250	161	23	203
84	66	87	4	37	110	80	230	215	42	243	245	81	240	248	175
190	196	239	97	212	115	205	92	141	156	129	176	72	167	184	63
171	52	221	104	228	28	232	164	38	195	24	9	226	133	217	113
143	54	128	53	15	16	253	90	125	211	18	112	222	60	105	172
102	178	186	109	67	47	146	51	8	220	173	140	107	45	252	137
132	88	152	185	22	29	157	62	43	3	189	247	210	183	49	20
163	127	134	100	95	174	59	158	108	79	213	7	2	204	124	251
187	197	238	149	169	64	138	117	236	136	77	89	131	198	165	119

Table 2. Nonlinearity scores of 8×8 S-Boxes

S-box	Nonlinearities										
	1	2	3	4	5	6	7	8	Min	Max	Mean
Proposed	106	108	108	106	104	106	104	106	104	108	106.0
Jakimoski *et al.* [11]	98	100	100	104	104	106	106	108	98	108	103.3
Chen *et al.* [12]	100	102	103	104	106	106	106	108	100	108	104.4
Asim *et al.* [13]	107	103	100	102	96	108	104	108	96	108	103.5
Wang *et al.* [15]	104	106	106	102	102	104	104	102	102	106	103.8
Özkaynak *et al.* [16]	104	100	106	102	104	102	104	104	100	104	103.3
Özkaynak *et al.* [17]	NA	NA	NA	NA	NA	NA	NA	NA	101	106	103.8
Skipjack S-box [29]	104	104	108	108	108	104	104	106	104	108	105.7
Xyi S-box [29]	106	104	104	106	104	106	104	106	104	106	105
Residue Prime [29]	94	100	104	104	102	100	98	94	94	104	99.5

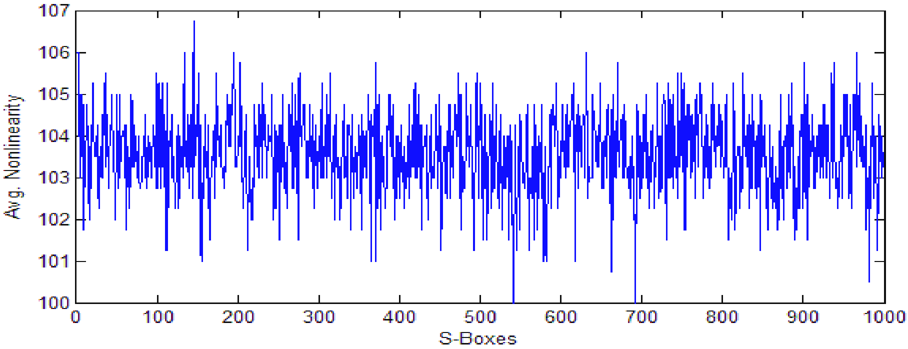


Fig. 2. Plot of average nonlinearities of 1000 S-boxes

Table 3. Dependency matrix of proposed S-Box

-	1	2	3	4	5	6	7	8
1	0.515	0.437	0.453	0.562	0.531	0.453	0.453	0.484
2	0.546	0.437	0.453	0.484	0.531	0.484	0.578	0.468
3	0.531	0.515	0.500	0.500	0.515	0.453	0.562	0.468
4	0.468	0.468	0.562	0.562	0.453	0.515	0.578	0.468
5	0.562	0.562	0.546	0.453	0.484	0.546	0.500	0.531
6	0.484	0.453	0.468	0.421	0.546	0.484	0.437	0.484
7	0.531	0.468	0.500	0.500	0.390	0.531	0.500	0.453
8	0.484	0.515	0.531	0.421	0.437	0.515	0.546	0.484

Table 4. Min-Max of dependency matrices and SACs of 8×8 S-Boxes

S-Box	Min	Max	SAC
Proposed	0.3906	0.5781	0.4965
Jakimoski *et al.* [11]	0.3750	0.5938	0.4972
Chen *et al.* [12]	0.4297	0.5703	0.4999
Asim *et al.* [13]	0.3906	0.5859	0.4938
Wang *et al.* [15]	0.4218	0.5681	0.4964
Özkaynak *et al.* [16]	0.4219	0.5938	0.5048
Özkaynak *et al.* [17]	0.4140	0.6328	0.5036
Skipjack S-box [29]	0.3750	0.6093	0.4980
Xyi S-box [29]	0.4218	0.5937	0.5048
Residue Prime [29]	0.4062	0.5937	0.5012

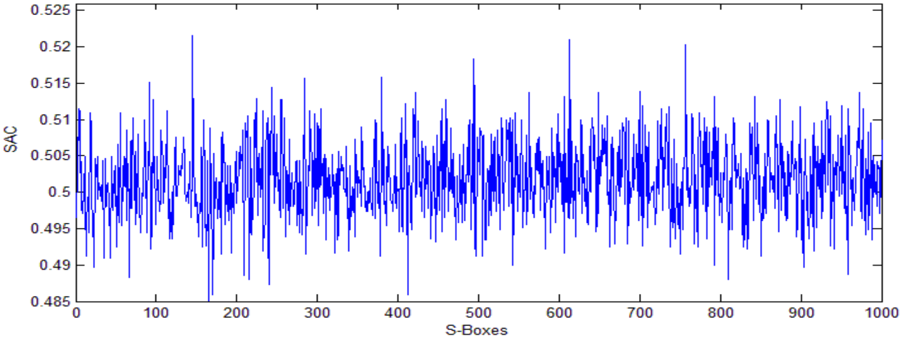


Fig. 3. Plot of SACs of 1000 S-boxes

0.4965 which is much close to the ideal value 0.5. The comparisons drawn in Table 4 highlight that the proposed S-box provides comparable parameter values with respect to the strict avalanche criteria. Moreover, the Fig. 3 shows that the SAC of all 1000 S-boxes lie within [0.485, 0.522] which is showing an excellent performance of proposed method with respect to the SAC property.

3.4 Equiprobable I/O XOR Distribution

In 1991, Biham and Shamir introduced the procedure of differential cryptanalysis to attack the DES-like cryptosystems [26]. They studied and exploited the imbalance on the input/output distribution. To resist the differential cryptanalysis, the XOR value of each output should have equal probability with the XOR value of each input. If an S-box is closed in I/O probability distribution, then it is resistant against differential cryptanalysis. It is desired that the largest value of *DP* should be as low as possible. The differential probability for a Boolean function *f*(x) is quantified as:

$$DP_f = \max_{\Delta x \neq 0, \Delta y} \left(\frac{\#\{x \in X \mid f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^n} \right)$$

Where *X* is the set of all possible input values and 2^n (here $n = 8$) is the number of its elements. The differential probabilities obtained for the proposed S-box are shown in Table 5. Now, it is found that the largest *DP* is 10 which is also the largest value in Asim's, Wang's and Özkaynak's S-boxes. However, this value is better than the Jakimoski's, Chen's, Skipjack's, Xyi's value of 12 and residue primes's of 72. This verifies that the proposed S-box is stronger than Jakimoski's, Chen's, Skipjack's, Xyi's, residue prime's S-boxes and comparable to others against differential cryptanalysis.

Table 5. Differential probabilities table in proposed S-Box

6	6	6	8	6	6	6	6	8	8	8	6	8	8	8	8
10	8	6	6	8	6	8	8	6	8	8	8	6	8	6	6
8	8	8	6	8	6	6	6	6	6	6	8	8	6	6	6
6	6	10	8	6	6	4	8	6	8	8	6	6	10	8	6
6	6	6	8	6	6	4	6	8	8	6	6	6	6	6	6
8	8	6	8	6	8	8	8	8	6	6	8	6	8	6	6
6	6	8	6	6	6	6	6	8	6	6	6	8	10	6	6
6	10	8	6	8	6	6	6	8	6	6	6	6	6	6	10
6	6	6	6	6	6	6	6	8	6	8	8	6	8	8	8
8	6	6	8	8	8	8	8	8	6	8	6	10	8	6	6
6	8	8	6	8	6	8	6	8	8	6	8	6	6	8	8
6	6	6	8	6	6	6	6	6	8	6	6	8	10	6	6
6	8	6	6	6	6	6	8	8	6	8	6	6	8	10	8
6	6	6	6	4	8	6	6	8	6	6	6	8	6	6	6
6	6	8	6	6	8	6	6	6	8	6	8	8	6	8	8
6	6	6	8	8	8	6	6	6	8	6	6	8	6	8	-

Table 6. Maximum differential probability of chaotic 8×8 S-Boxes

S-Box	Max DP
Proposed	10/256
Jakimoski *et al.* [11]	12/256
Chen *et al.* [12]	12/256
Asim *et al.* [13]	10/256
Wang *et al.* [15]	10/256
Özkaynak *et al.* [16]	10/256
Özkaynak *et al.* [17]	10/256
Skipjack S-box [29]	12/256
Xyi S-box [29]	12/256
Residue Prime [29]	72/256

4 Conclusion

A novel effective method is proposed to construct efficient cryptographic substitution-boxes. The method is based on the classical Fisher-Yates shuffle algorithm. The piece-wise linear chaotic map is employed to enhance the effectiveness of the shuffle algorithm. The generation of dynamic S-boxes is under the control of secret key. The simplicity and consistency are the key features of the proposed methodology. Moreover, a number of different S-boxes can be easily generated. The statistical analyses show that the proposed method is credential of generating efficient dynamic S-boxes and verify its practicableness as nonlinear components in the design of strong block encryption systems.

References

1. Shannon, C.E.: Communication theory of secrecy systems. Bell Systems Technical Journal 28, 656–715 (1949)
2. Menezes, A.J., Oorschot, P.C.V., Vanstone, S.A.: Handbook of applied cryptography. CRC Press (1997)
3. Ahmad, M.: Design and FPGA implementation of LFSR based data encryption circuit for secure communication, M.Tech Dissertation, AMU Aligarh (2008)
4. Dawson, M.H., Tavares, S.E.: An expanded set of S-box design criteria based on information theory and its relation to differential-like attacks. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 352–367. Springer, Heidelberg (1991)
5. Chen, G.: A novel heuristic method for obtaining S-boxes. Chaos, Solitons & Fractals 36(4), 1028–1036 (2008)
6. Karaahmetoglu, O., Sakalli, M.T., Bulus, E., Tutanescu, I.: A new method to determine algebraic expression of power mapping based S-boxes. Information Processing Letters 113(7), 229–235 (2013)
7. Szaban, M., Seredynski, F.: Designing cryptographically strong S-boxes with the use of cellular automata. Annales UMCS Informatica Lublin-Polonia Sectio AI 8(2), 27–41 (2008)
8. Cusick, T.W., Stanica, P.: Cryptographic Boolean Functions and Applications. Elsevier, Amsterdam (2009)
9. Youssef, A.M., Tavares, S.E., Gong, G.: On some probabilistic approximations for AES-like S-boxes. Discrete Mathematics 306(16), 2016–2020 (2006)
10. Bard, G.V.: Algebraic Cryptanalysis. Springer, Berlin (2009)
11. Jakimoski, G., Kocarev, L.: Chaos and cryptography: Block encryption ciphers based on chaotic maps. IEEE Transaction on Circuits Systems 48(2), 163–169 (2001)
12. Chen, G., Chen, Y., Liao, X.: An extended method for obtaining S-boxes based on three-dimensional chaotic Baker maps. Chaos, Solitons & Fractals 31(3), 571–577 (2007)
13. Asim, M., Jeoti, V.: Efficient and simple method for designing chaotic S-boxes. ETRI Journal 30(1), 170–172 (2008)
14. Yin, R., Yuan, J., Wang, J., Shan, X., Wang, X.: Designing key-dependent chaotic S-box with large key space. Chaos, Solitons & Fractals 42(4), 2582–2589 (2009)
15. Wang, Y., Wong, K.W., Liao, X., Xiang, T.: A block cipher with dynamic S-boxes based on tent map. Communications in Nonlinear Science and Numerical Simulations 14(7), 3089–3099 (2009)

16. Özkaynak, F., Özer, A.B.: A method for designing strong S-boxes based on chaotic Lorenz system. Physics Letters A 374(36), 3733–3738 (2010)
17. Özkaynak, F., Yavuz, S.: Designing chaotic S-boxes based on time-delay chaotic system. Nonlinear Dynamics 74(3), 551–557 (2013)
18. Fisher, R.A., Yates, F.: Statistical tables for biological, agricultural and medical research, 3rd edn., pp. 26–27. London Oliver & Boyd. (1938)
19. Durstenfeld, R.: Algorithm 235: Random permutation. Communications of the ACM 7(7), 420 (1964)
20. Knuth, D.E.: Seminumerical algorithms. In: The Art of Computer Programming, vol. 2, pp. 124–125. Addison–Wesley (1969)
21. Li, S., Chen, G., Mou, X.: On the dynamical degradation of digital piecewise linear chaotic maps. International Journal of Bifurcation and Chaos 15(10), 3119–3151 (2005)
22. Liu, Y., Tong, X.J.: A new pseudorandom number generator based on complex number chaotic equation. Chinese Physics B 21(9), 90506–90508 (2012)
23. Alvarez, G., Li, S.: Some basic cryptographic requirements for chaos-based cryptosystems. International Journal of Bifurcation and Chaos 16(8), 2129 (2006)
24. Hermassi, H., Rhouma, R., Belghith, S.: Improvement of an image encryption algorithm based on hyper-chaos. Telecommunication Systems 52(2), 539–549 (2013)
25. Kanso, A., Yahyaoui, H., Almulla, M.: Keyed hash function based on a chaotic map. Information Sciences 186(1), 249–264 (2012)
26. Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology 4(1), 3–72 (1991)
27. Dalai, D.K.: On some necessary conditions of boolean functions to resist algebraic attacks, PhD thesis, ISI Kolkata (2006)
28. Webster, A.F., Tavares, S.E.: On the design of S-boxes. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 523–534. Springer, Heidelberg (1986)
29. Hussain, I., Shah, T., Gondal, M.A., Khan, W.A.: Construction of cryptographically strong 8x8 S-boxes. World Applied Sciences Journal 13(11), 2389–2395 (2011)

Experimental Analysis on Access Control Using Trust Parameter for Social Network

Saumya Omanakuttan and Madhumita Chatterjee

Pillai's Institute of Information Technology, Navi Mumbai, India
o.saumya@gmail.com, mchatterjeee@mes.ac.in

Abstract. Technology made socializing very simple and easy, connecting everyone is just a matter of a click today. The security of our personal information and sharing that information in the digital world has always been a major challenge for the ever-growing social networks. When it comes to the relationship between people and technology, the attribution of trust is a matter of dispute always. This paper proposes an access control scheme called Trust Based Access Control for Social Networks, or STBAC, which allows users to share data among their friends, using a trust computation to determine which friends should be given access. This trust computation uses previous interactions among a user's friends to classify his or her peers into privileged or unprivileged zones, which determine whether that peer gains access to the user's data. The system will work as a filter for each of the peer and try to evaluate the trust access control in social networks.

Keywords: Trust, credibility, reliability, PeerTrust, transaction.

1 Introduction

Man is a social animal and the ever growing social networking sites just proves it. One can post their idea, share their picture, homemade video etc. without which now life seems to be very incomplete, since it helps ones to catch up with daily update of the dear ones with the hectic schedule. Together with such a fast spreading activity, various concerns and risks become obvious. The establishment of trust and the protection of users becomes an ongoing challenge within the online social networking environment, with the threat of misuse and privacy intrusions by malicious users illustrating this challenge. The vast pool of friends with includes friends and virtual friend is a matter of concern for many treasured information which is shared on the social networks may be with or without user consent.

2 Related Work

There are many related work on access control to make sure that the social network to be secure enough. Peer – to- peer online communities which offers threats and opportunities, different model to evaluate trust in a decentralized environment to evaluate the effectively PeerTrust Model [1 2] for ecommerce communities.

G. Martínez Pérez et al. (Eds.): SNDS 2014, CCIS 420, pp. 551–562, 2014.

The research about access control based trust promotes development of access control to use trust as a parameter.. Thus design a trust-based access control (TBAC) [3] with feedback. Trust into role based access control model (TRBAC) [4], they proposed an infrastructure-centric enforcement framework. With the RBAC extension, trust level requirements, which dictate the trust prerequisites for role activation under the privilege context, can be specified.. The created platform was based on the extended RBAC model that provides the developers more flexibility and complex vie of security organization [5].

In [7], Trust in an identity and its associated profile attributes is generally intended as a prerequisite for a secure determination of entitlements. The NIST model seeks to resolve this situation by unifying ideas from prior RBAC models [8], commercial products and research prototypes. It is intended to serve as a foundation for developing future standard. RBAC is a rich and open-ended technology. The idea of current social networks implements very simple protection mechanisms [9], with this aim, they proposed an access control model where authorized users are denoted based on the relationships they participate in.

Service providers are focused primarily on acquiring users and little attention is given to the effective management of these users within the social networking environment [10]. In [11], a mechanism for accessing data containing personally identifiable information, the key component of the framework is an access control model that provides full support for expressing highly complex privacy related policies, taking into account features like purposes and obligations. In [12], practical solution that establishes a trust infrastructure, enables authenticated and secure communication between users in the social network and provides personalized, fine grained data access control.

3 Proposed System

The proposed system a “Trust Based Access Control for Social Networks” (STBAC), which allows users to share data among their friends, using a trust computation to determine which friends should be given access. Just as in real life relationships, the trust levels can vary from friend to friend, and may change over time.

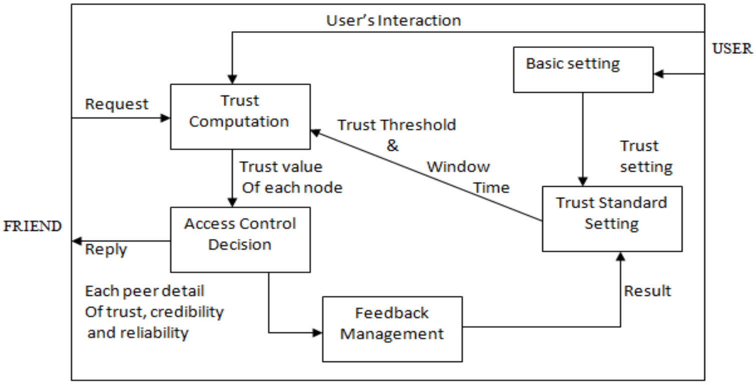


Fig. 1. The Proposed System for Trust based Access Control for Social Networks (STBAC) [1]

The STBAC is used as a filter in order divide the friends into two broad category of privileged and unprivileged friends where a privileged friend can view all the detail of the user like wall, photos, personal information and can share as well but the unprivileged friend won't be able to view users photos or video as per the user request. This filtration is based on a major concern human parameter, Trust. The figure 1 describes the proposed system for social networks which consist of two actors involved the user and friend. The entire system consist of major five modules such as basic setting, trust standard setting, feedback management, access control decision and the trust computation. The detail working of the proposed system is explained in [1].

4 Trust Calculation and Algorithm for Social Networks

4.1 Proposed STBAC Calculation

In a social network trust plays a very vital role. The trust of each peer is calculated on the bases of number of transaction being performed with the neighbouring peer or the friend involved in the network. Suppose 'X' is user who has 'n' number of friends thus X will have transaction such as comments, like, message or scrap, sharing of images etc with these nodes.

Credibility of a Peer : Credibility of the peer is defined as the summation of Boolean value of difference between the incoming and outgoing transaction of the user with its peer friends. Suppose a peer 'u' has 'i' peer as friends or the neighboring peer. Thus performs 'X 'transaction. The incoming transaction is denoted as '$X_{incoming}$' and outgoing transaction with any peer is denoted as '$X_{outgoing}$'. The difference between the incoming and outgoing transaction is denoted as'd'.

$$d= X_{incoming} - X_{outgoing} \tag{1}$$

$$\begin{array}{rl} \text{if} \quad d > 0 & \|\ 1 \text{ denoted as dp} \\ \text{Else if} \quad d = 0 & \|\ 1 \text{ denoted as do} \\ \text{Otherwise} & \|0 \text{ denoted as dn} \end{array}$$

If d is a positive value implies that the incoming transaction is more than the outgoing transaction of node 'u' thus denoted as 'dp'

If d is a negative value implies the incoming transaction is less than the outgoing transaction of a node 'u' thus denoted as 'dn '. If d is zero which implies two conditions that either the incoming and outgoing transactions are equal or there is no transaction at all within the peers denoted as 'do'.

Thus credibility (Cr) of a peer is defined as

$$Cr\ (p\ (u, i)) = \{ \tfrac{1}{i} \{ \sum dp\ + \sum do \} \} * 100 \tag{2}$$

$$\begin{array}{lll} \text{if} & Cr\ (p\ (u, i)) >= 70 & \|\ 1 \\ \text{Else if} & Cr\ (p\ (u, i)) > 50 \text{ and } < 70 & \|0.5 \\ & \text{Otherwise} & \|\ 0 \end{array}$$

If the Cr(p(u)) is greater than or equal 75% implies than the friend node ‘i’ assures that the user ‘u’ has high credibility thus is assigned ‘1’

If the Cr (p (u)) is less than 70% and more than or equal to 50% implies than the friend node ‘i‘assures that the user ‘u’ has low credibility thus assigned ‘0.5’.

If the Cr (p (u)) is less than 50% implies than the friend node ‘i’ assures that the user ‘u’ has low credibility thus assigned ‘0’.

Direct Trust : Direct Trust of a peer is determined on the basis of total number of transaction performed between the user and its peer friends. Suppose ‘u’ has n number of transaction with another peer ‘v’. Thus trust (T) of the peer ‘u’ with respect to ‘v’ will be calculated as follows.

$$\text{T(p(u,v)} = \frac{\text{Xoutgoing (p(v))}}{\sum p(u,i)} * 100 \tag{3}$$

T(p(u,v) > = TTh || Notify Allow access to be authorized friend

Otherwise || Access denied, remain as unauthorized friend

Where

p (u,v) = transaction between the user ‘u’ and a friend node ‘v’

p (u,i) = total number of transaction of u with all the friend node ‘i’ in the social network

p (v) = friend node ‘v’

X_{outgoing} = Outgoing transaction

T_{Th} = Trust Threshold which is input as per the user.

If the trust value of p(u,v) is greater than the set T_{Th} then the friend node ‘v’ should be notified that the friend who is unauthorized, should be granted to become an authorized friend if accepted will be given all the rights of authorized to check the scrap, photos etc.else will be reject to remain the unauthorized friend.

Reliability of a Node: Reliability of a peer is defined as the summation of product of the credibility of the friend peer and direct trust of user in the networks denoted as R(u,i) where u is the host whose reliability is determined with respect to the friend peer ‘i’ in the network thus calculated as follows.

$$\text{R(u, i)} = \sum_{i=1}^{N} T(p(u,i)) * Cr(i) \tag{4}$$

Where T(p(u,i)) is direct trust value of user and Cr(i) is the credibility of a friend peer. Thus we can check the reliability of a user thus results a reduction in the dummy node.

PeerTrust of a node: PeerTrust of a node is defined as the summation of product of the credibility of the friend peer and the incoming transaction of the friend peer ‘i’ to user in the network denoted as PT(u,i) where ‘u’ is the host or the source node whose PeerTrust is to be calculated with to respect to the friend peer ’i’ in the network is calculated as follows.

$$PT(u,i)=1/n*\sum_{i=0}^{n}(Cr(i) * Incoming(p(u,i))) \quad (5)$$

Where Incoming(p(u,i)) is the incoming value of the friend peer and Cr(i) is the credibility of a peer friend. Thus we can check the PeerTrust of a user. Thus the above four trust parameter is purely time dependent as per the user interest. Thus the evaluation of the system will result in the better access system for social networks.

4.2 Proposed Algorithm for STBAC

```
Input:$u$                    // $u$ is the user
/* Set the trust threshold of each user $T_{Th}$ (u) and the window period */
for u=1 to n do
        Set ( $u$, win)
          $T_{Th}$(u) -------- Trust Threshold                // set the trust value
end for
/*$i$ is all friend peer with whom node $u$ is interacting for ‘win’ window period, thus ranging from 1 to n node in the friend list */
//   ‘$v$’ is transaction performed for a single friend peer

for $i$= 1 to n
     for v = 1 to n
         d(p($u$,$i$)) <= feedback source ($v$) using Eq.1

         if d(p($u$,$i$)) < 0 then
              d(p($u$,$i$)<= bool(0)
     else
              d(p($u$,$i$)<=bool(1)
     end if
         Cr(p($u$,i)) <= Calculate the credibility Eq. 2
     end for
end for

/* Reliability of the node is calculate for each user
for u=1 to n
         R(p(u,i)) <= Calculate the reliability Eq.4
end for
/* PeerTrust of the node is calculated for each user
for u=1 to n
         PT(p(u,i)) <= Calculate the PeerTrust Equation Eq.5
end for
```

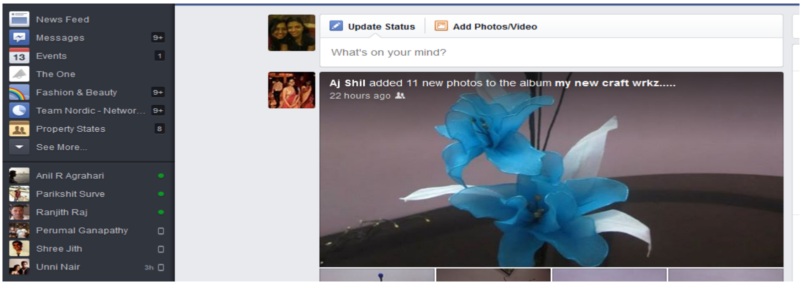
```
// Feedback Management
for $i$=1 to n do
    Feedback ( $i$, $win$)          /* ‘$i$’ is peer with whom node u is
                                       interacting for ‘$win$’ window period*/
      T(p($u$,$i$))                 // trust computation using Eq. 3
      Ts(p($u$,$i$))                // New trust value computed after
                                         the   window period
        if  Ts(p($u$,$i$)) > $T_{Th}$ then
             feedback($i$) <= grant access, to be authorized user
        else
             feedback($i$) <= reject, continue as unauthorized user
        end if

     Feedback ($i$,$win$)
     count($i$)
end for
```

5 Comparative Study

On the basis of the various parameters such as display of homepage, groups formation, detection of dumpy node etc.., the existing system is compared with proposed system is explained below.

5.1 Display of Home Page

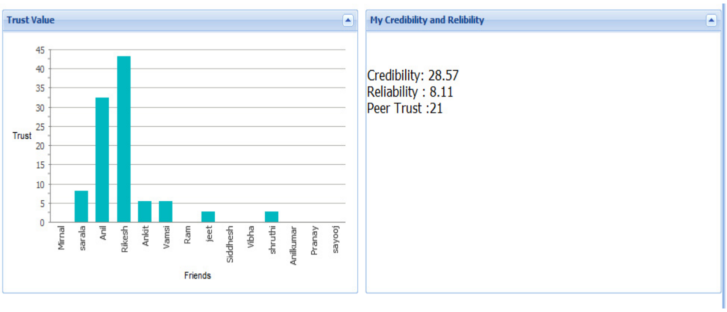
The home page of existing system of the social network displays the detail of the peer information i.e the activity being performed by the friends of the user



Aj Shil added 11 new photos to the album my new craft wrkz.....
22 hours ago

Fig. 2. The existing system home page

The proposed system Displays the detail of transaction with each friends thus estimate the parameter like Trust, Credibility, reliability and PeerTrust.



Credibility: 28.57
Reliability : 8.11
Peer Trust :21

Fig. 3. The home page of the proposed system

5.2 Categorization of Friends(Static /Dynamic)

In the existing system the user need to categorize the friend in static group and they remain so throughout provided user don't explicitly change the group. The groups are given static privacy setting as per standard question.

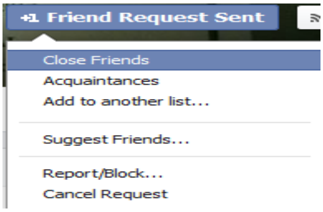
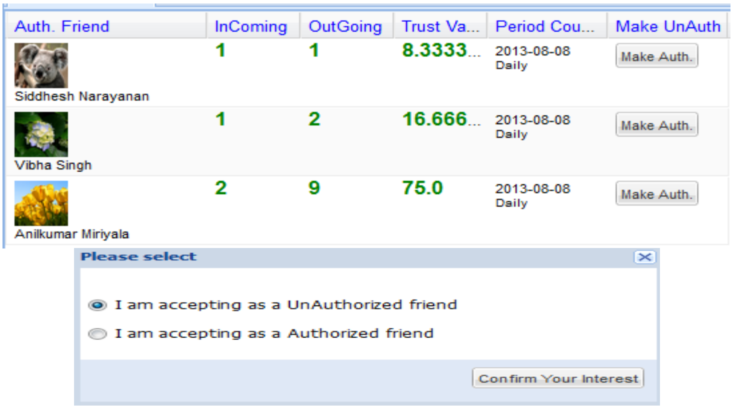


Fig. 4. Existing system the friends are divided into various group as close friends, acquaintances etc.

In the proposed system there are only two groups authorized or unauthorized. Depending on The direct trust value and the threshold value the user are categorized into groups. Since the direct trust value depends on the transaction of the user with its friends with respect to time it is dynamic in nature.



Auth. Friend	InComing	OutGoing	Trust Va...	Period Cou...	Make UnAuth
Siddhesh Narayanan	1	1	8.3333...	2013-08-08 Daily	Make Auth.
Vibha Singh	1	2	16.666...	2013-08-08 Daily	Make Auth.
Anilkumar Miriyala	2	9	75.0	2013-08-08 Daily	Make Auth.

Please select

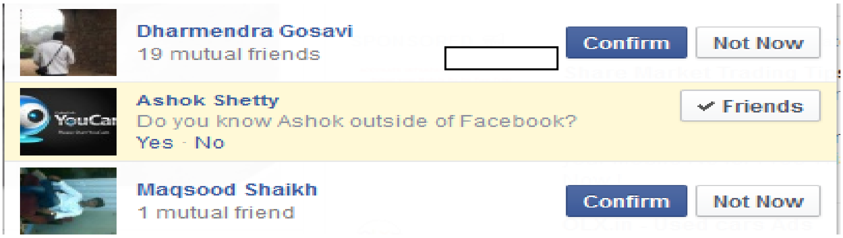
● I am accepting as a UnAuthorized friend
○ I am accepting as a Authorized friend

Confirm Your Interest

Fig. 5. Proposed System, friends are divided into two group depending on direct trust value

5.3 Recognition of Dummy Node

In existing system the dummy node is evaluated on the basis of mutual friend or whether user know the person outside the social network



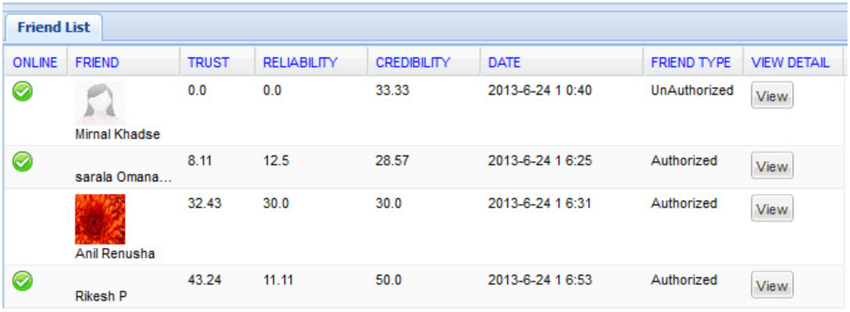
Dharmendra Gosavi
19 mutual friends

Ashok Shetty
Do you know Ashok outside of Facebook?
Yes · No

Maqsood Shaikh
1 mutual friend

Fig. 6. Recognition of dummy node is based on mutual friend and whether the user know the friend outside social network

In proposed system the dummy node is evaluated on various parameter which is displayed to the user in friend list. The various parameter through which a user can estimate the dummy node are credibility, reliability etc.



Friend List

ONLINE	FRIEND	TRUST	RELIABILITY	CREDIBILITY	DATE	FRIEND TYPE	VIEW DETAIL
✓	Mirnal Khadse	0.0	0.0	33.33	2013-6-24 1 0:40	UnAuthorized	View
✓	sarala Omana...	8.11	12.5	28.57	2013-6-24 1 6:25	Authorized	View
	Anil Renusha	32.43	30.0	30.0	2013-6-24 1 6:31	Authorized	View
✓	Rikesh P	43.24	11.11	50.0	2013-6-24 1 6:53	Authorized	View

Fig. 7. Recognition of dummy node is based various trust parameter

6 Experimental Analysis and Results

6.1 Analysis of Minimizing the Dummy Node

The proposed system have ‘n’ number of user for the social network who are interacting with ‘n’ peers. For evaluating the result some sample user are considered, suppose there are seven login user such as Shruthi, Anil, Sayooj, Saumya, Vibha, Pranay and siddhesh, having following as the Trust Parameter. The data which is generated in table 1 are being calculated using the formula mentioned in section 4 depending of the login user transaction with each of its friend peer.

Table 1. Original value of the Trust Parameter

Name of the Peer	Credibility (in %)	Reliability (in %)	PeerTrust (in %)
Shruthi	71.43	5.56	35
Anil	50	52.62	50
Sayooj	83.33	20	40.5
Saumya	28.57	2.56	16.5
Vibha	33.33	44.44	33.33
Siddhesh	66.67	33.33	50
Pranay	83.33	25	41.67

Table 2. Observation for the Trust Parameter

	Credibility >=70	>=50 and <70	<50
Reliability	Increases	Decreases	Decreases
Peertrust	Increase by 1	Increase by 0.5	No impact

If the credibility greater than or equal to 70 then any peers having a transaction with such a peer will increase its reliability. The peer with high credibility when interact with any other peer will result in the increase of PeerTrust value of the peer. If the credibility greater than or equal to 50 but less than 70 then having transaction with such a peer will decrease the reliability but if these peer have a transaction with any other peer will contribute 0.5 increase in the PeerTrust value of the that peer. If the credibility is less than 50 then either the peer is new or may be malicious, thus decreases the trust parameter.

Table 3. Observation for Reliability

Login User	Friend with (credibility)	Initial Reliability (%)	After The transaction Reliability (%)	Result
Sayooj	Shruthi (42.86)	20	16.67	Decrease
Sayooj	Pranay (83.33)	16.67	28.58	Increase
Sayooj	Saumya (28.57)	28.58	20	Decrease
Sayooj	Anil (50)	20	18.18	Decrease

According to Table 3, the reliability of the peer only increases provide the login user have the transaction with the credential high peer otherwise the reliability of the peer decreases. Thus the reliability of the peer evaluates or indicates whether the login user has transactions with high credential peer or not. If value of reliability is zero indicate the login user never had a conversation/transaction with any peer.

Table 4. Observation for PeerTrust

LoginUser	Friend with (credibility in %)	Total No. of Friend	PeerTrust (%)

Sayooj	Shruthi(42.86)
	Anil (66.67)
	Saumya(35.71)
	Pranay(83.33)
		6	25

Anil	Sayooj(83.33)
	Siddesh(66.67)
	Pranay(83.33)
	Vibha(66.67)
		6	50

Shruthi	Siddesh(83.33)
	Anil(66.67)
	Vibha(33.33)
	Pranay(83.33)
	Syooj(83.33)
	Saumya(35.71)
	Payal(10.0)
		7	50

According to Table 4, it is observed that the total number of user interaction with the login user is estimated for the PeerTrust evaluation. Thus the peer with high credential value will increase the PeerTrust value by 1, medium credential value will increase the login user peer value by 0.5, and low credential value will contribute for nothing. If the Value of PeerTrust is 100% that mean all the peer which belong to login user all are credential high peer and they are conversing regularly whereas if PeerTrust is zero that indicate that either the login user is new or none of the credential high peer are part of the login user or high credential peer do not respond the login user, this also indicate that may be the peer is malicious, thus can minimize the dummy node.

Thus the above calculation is true for all the peer in the social network. As the number of friend peer increases in the social network. The calculation of each parameter of trust module will be fine and precise thus helping the user to understand and verify the dummy nodes.

6.2 Result to Categorize the Friends into Authorized and Unauthorized Category Dynamically

Suppose if the user want to secure its own account privacy is done on the basses of trust threshold value which depends on transaction by the user with its respective friends to the total number of transaction performed by the user according to the set time span. For the privacy setting of user account, a trust threshold value and duration which set by the user and the corresponding trust value of the user will be calculated and if the user cross the threshold value set by the user then the report will be generated in order to categories the user into the authorized category of friends where he/she will get the access right to check everything of the user or will they be unauthorized so that the user can prevent their account from the unwanted friend or malicious user. Thus the table 5 illustrates three users Vibha, Anil and Saumya with respective thresholds and transaction.

Table 5. Observation for Privacy Setting

Login user	Threshold (Th)	Friend	No.of transaction	Total no.of transaction
			outgoing	
Vibha	Trust Th =60 Duration= weekly	Anil	20	100
		Sayooj	75	
		Shruthi	5	
Anil	Trust Th =20 Duration = monthly	Vibha	20	140
		Saumya	80	
		Pranay	40	
Saumya	Trust Th=90 Duration = daily	Siddesh	1	40
		Vibha	37	
		Anil	2	

As per table 6 the report will be generated for the entire friend peer who crosses the trust threshold value after the specified duration of time. The report will ask the user for permission to whether grant access to the friend peer for authorization or can the decline friend peer access.

Table 6. Observation for Trust Access Report Generation

Login user	Threshold (Th)	Friend	Trust Value (T(u)) (%)	Report generation T(u)>=(T_{TH})
Vibha	Trust Th = 60 Duration = weekly	Anil	20	NO
		Sayooj	75	YES
		Shruthi	5	NO
Anil	Trust Th = 20 Duration = monthly	Vibha	14.28	NO
		Saumya	57	YES
		Pranay	28.57	YES
Saumya	Trust Th= 90 Duration = daily	Siddesh	2.5	NO
		Vibha	92.5	YES
		Anil	5	NO

Since it is time dependent user can keep the track of its entire authorized user and change accordingly as per their trust value and behavior at any point of time. Thus reducing the issue related with privacy constraint of the user which is not static as in case of existing system but dynamic since it depends on time.

7 Future Scope and Conclusion

The proposed trust based access control for social network (STBAC) allows user to differentiate among his or her friends in the social network, dynamically. STBAC also help user to identify the malicious peer via a credibility and reliability and PeerTrust parameter for each peer, thus reducing the dummy node. In future to increase the efficiency of the system a penalty module can be included to make the trust parameter more dynamic and which can help in more precise trust parameter. To have a more granular form of trust model similarity algorithms can also be used. As social networks become more popular, they will become an increasingly important method of communication. Because of this, it is of vital importance that we start considering effective and flexible access control scheme to protect the data in social network.

References

1. Omanakuttan, S., Chatterjee, M.: Trust Based Access Control for Social Networks(STBAC). International Journal of Innovations in Engineering and Technology (IJIET), 325–331 (February 1, 2013)
2. Li, X., Ling, L.: PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities. IEEE Transactions on Knowledge and Data Engineering 16(7), 843–857 (2004)
3. Ma, X., Feng, Z., Xu, C., Wang, J.: A Trust-Based Access Control with Feedback. In: 2008 International Symposiums on Information Processing (ISIP), May 23-25, pp. 510–514 (2008)
4. Yang, R., Lin, C., Jiang, Y., Chu, X.: Trust Based Access Control in Infrastructure-Centric Environment. In: 2011 IEEE International Conference on Communications (ICC), June 5-9, pp. 1–5 (2011)
5. Poniszewska-Maranda, A.: Platform for Access Control Management in Information System Based on Extended RBAC Model. In: 2010 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp. 510–517 (September 2010)
6. Lampson, B.: Protection. In: Proceedings of the 5th Annual Princeton Conference on Information Sciences and Systems, pp. 437–443. Princeton University (1971)
7. Benantar, M.: Access Control Systems: Security, Identity Management and Trust Models. Springer, New York (2006)
8. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computer 29(2), 38–47 (1996)
9. Carminati, B., Ferrari, E., Perego, A.: Rule-based access control for social Networks. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops, Part II. LNCS, vol. 4278, pp. 1734–1744. Springer, Heidelberg (2006)
10. Galpin, R., Flowerday, S.V.: Online social networks: Enhancing user trust through effective controls and identity management. In: Information Security South Africa (ISSA), August 15-17, pp. 1–8 (2011)
11. Wang, H., Sun, L.: Trust-Involved Access Control in Collaborative Open SociaNetworks. In: 2010 4th International Conference on Network and System Security (NSS), September 1-3, pp. 239–246 (2010)
12. Graffi, K., Mukherjee, P., Menges, B., Hartung, D., Kovacevic, A., Steinmetz, R.: Practical security in p2p- based social networks. In: IEEE 34th Conference on Local Computer Networks, LCN 2009, October 20-23, pp. 269–272 (2009)

Author Index