

Jiankun Hu · Ibrahim Khalil  
Zahir Tari · Sheng Wen (Eds.)



235

LNICST

# Mobile Networks and Management

9th International Conference, MONAMI 2017  
Melbourne, Australia, December 13–15, 2017  
Proceedings



# **Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering**

**235**

## **Editorial Board**

Ozgur Akan

*Middle East Technical University, Ankara, Turkey*

Paolo Bellavista

*University of Bologna, Bologna, Italy*

Jiannong Cao

*Hong Kong Polytechnic University, Hong Kong, Hong Kong*

Geoffrey Coulson

*Lancaster University, Lancaster, UK*

Falko Dressler

*University of Erlangen, Erlangen, Germany*

Domenico Ferrari

*Università Cattolica Piacenza, Piacenza, Italy*

Mario Gerla

*UCLA, Los Angeles, USA*

Hisashi Kobayashi

*Princeton University, Princeton, USA*

Sergio Palazzo

*University of Catania, Catania, Italy*

Sartaj Sahni

*University of Florida, Florida, USA*

Xuemin Sherman Shen

*University of Waterloo, Waterloo, Canada*

Mircea Stan

*University of Virginia, Charlottesville, USA*

Jia Xiaohua

*City University of Hong Kong, Kowloon, Hong Kong*

Albert Y. Zomaya

*University of Sydney, Sydney, Australia*

More information about this series at <http://www.springer.com/series/8197>

Jiankun Hu · Ibrahim Khalil  
Zahir Tari · Sheng Wen (Eds.)

# Mobile Networks and Management

9th International Conference, MONAMI 2017  
Melbourne, Australia, December 13–15, 2017  
Proceedings



Springer

*Editors*

Jiankun Hu  
SEIT, UNSW Canberra  
Canberra  
Australia

Ibrahim Khalil  
RMIT University  
Melbourne, VIC  
Australia

Zahir Tari  
RMIT University  
Melbourne, VIC  
Australia

Sheng Wen  
Swinburne University of Technology  
Hawthorne, Melbourne, VIC  
Australia

ISSN 1867-8211

ISSN 1867-822X (electronic)

Lecture Notes of the Institute for Computer Sciences, Social Informatics  
and Telecommunications Engineering

ISBN 978-3-319-90774-1 ISBN 978-3-319-90775-8 (eBook)

<https://doi.org/10.1007/978-3-319-90775-8>

Library of Congress Control Number: 2018941835

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2018  
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

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

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

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG  
part of Springer Nature  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

## Preface

We are delighted to introduce the proceedings of the 9th European Alliance for Innovation (EAI) International Conference on Mobile Networks and Management (MONAMI 2017). This conference has brought together researchers, developers, and practitioners from around the world who are leveraging and developing mobile computing, wireless networking, and management.

The technical program of MONAMI 2017 consisted of 30 full papers in the main conference track. The conference had a special session on Trust, Privacy and Security in Internet of Things (IoT) and Cloud. Aside from the high-quality technical paper presentations, the technical program also featured five keynote speeches. These keynote speakers were Prof. Wanlei Zhou from Deakin University Australia, Prof. Jie Lu from University of Technology Sydney Australia, Prof. Joe Dong from UNSW Sydney Australia, Prof. Yongsheng Gao from Griffith University Australia, and Dr. Shui Yu from Deakin University.

Coordination with the steering chairs, Dr. Imrich Chlamtac from Create-Net, Italy, and Prof. Jiankun Hu and Conference General Co-Chair Prof. Yang Xiang from Swinburne University Australia from UNSW Canberra, Australia, was essential for the success of the conference. We sincerely appreciate their constant support and guidance. It was also a great pleasure to work with such an excellent Organizing Committee who worked hard in organizing and supporting the conference. In particular, the Technical Program Committee, led by our TPC co-chairs, Dr. Zahir Tari and Dr. Ibrahim Khalil from RMIT Australia, completed the peer-review process for the technical papers and compiled a high-quality technical program. We are also grateful to other chairs including the publication chair, Dr. Sheng Wen from Swinburne University of Technology, the PC members for their support, and all the authors who submitted their papers to the MONAMI 2017 conference and workshops.

We strongly believe that MONAMI 2017 provided a good forum for all researchers, developers, and practitioners to discuss all scientific and technological aspects that are relevant to mobile networks and management. We also expect that future MONAMI conferences will continue being successful and stimulating, as indicated by the contributions presented in this volume.

April 2018

Jiankun Hu  
Ibrahim Khalil  
Zahir Tari

## Conference Organization

## **Steering Committee**

Imrich Chlamtac (Chair) EAI/CREATE-NET, Italy  
Jiankun Hu UNSW Canberra, Australia

## **Organizing Committee**

## **General Co-chairs**

Jiankun Hu UNSW Canberra, Australia  
Xiang Yang Deakin University, Melbourne, Australia

## Technical Program Committee Chairs and Co-chairs

Zahir Tari RMIT University, Melbourne, Australia  
Ibrahim Khalil RMIT University, Melbourne, Australia

Web Chair

Naveen Chilamkurti Latrobe University, Melbourne, Australia

### **Publicity and Social Media Chair/Co-chair**

Chan Yeob Yeun Khalifa University, UAE

Workshops Chair

Ibrahim Khalil RMIT University, Melbourne, Australia

## **Sponsorship and Exhibits Chair**

Jiankun Hu UNSW, Australia

## **Publications Chair**

Sheng Wen Deakin University, Australia

Panels Chair

Jiankun Hu UNSW, Australia

Tutorials Chair

Ibrahim Khalil RMIT University, Melbourne, Australia

**Demos Chair**

Ibrahim Khalil RMIT University, Melbourne, Australia

**Posters and PhD Track Chair**

Ibrahim Khalil RMIT University, Melbourne, Australia

**Local Chair**

Ibrahim Khalil RMIT University, Melbourne, Australia

**Honorary Co-chairs**

Albert Zomaya	The University of Sydney, Australia
Mohammed Atiquzzaman	University of Oklahoma, USA
Hsiao-Hua Chen	National Cheng Kung University, Tainan
Ernesto Damiani	Khalifa University, UAE

**Conference Manager**

Alzbeta Mackova EAI (European Alliance for Innovation)

**Technical Program Committee**

Neda Abutorab	UNSW, Australia
Adnan Al-Anbuky	AUT, New Zealand
Abderrahim Benslimane	University of Avignon, France
Abbas Bradai	University of Poitiers, France
Claudia Canali	UNI More, Italy
Chi Chen	Chinese Academic of Science, China
Naveen Chilamkurti	La Trobe University, Australia
Baldomero Coll-Perales	UWICORE, Israel
Antonio De Domenico	CEA, France
Mounia Fredj	ENSIAS, France
Ying Guo	Central South University, China
Song Guo	The Hong Kong Polytechnic University, SAR China
Slimane Hammoudi	Université d'Angers, France
Jiankun Hu	UNSW, Australia
Xinyi Huang	Fujian Normal University, China
Qin Jing	Shandong University, China
Ibrahim Khalil	RMIT University, Australia
Weifa Liang	Australia National University, Australia
Qin Liu	Hunan University, China
Constandinos Mavromoustakis	Cyprus
Emad Eldin Mohamed	Swinburne University, Australia
Miklos Molnar	University of Montpellier 2, France

Paul Pan	Unitec, New Zealand
Yi Qian	University of Nebraska – Lincoln, USA
Abderrezak Rachedi	Université Paris-Est Marne-la-Vallée, France
Javier Rubio-Loyola	Mexico Teléfono, Mexico
Henning Sanneck	Nokia, Germany
Zhili Sun	University of Surrey, UK
Zahir Tari	RMIT University, Australia
Abdelkamel Tari	University of Bejaia, Algeria
Craig Valli	ECU, Australia
Anna Maria Vegni	University of Rome, Italy
Sheng Wen	Deakin University, Australia
Qianghong Wu	Beihang University, China
Chan Yeob Yeun	Khalifa University, UAE
Arkady Zaslavsky	Data61, Australia
Sherali Zeadally	University of Kentucky, USA
RongBo Zhu	South-Central University for Nationalities, China

# Contents

Offloading of Fog Data Networks with Network Coded Cooperative D2D Communications . . . . .	1
<i>Ben Quinton and Neda Abutorab</i>	
Persistent vs Service IDs in Android: Session Fingerprinting from Apps . . . . .	14
<i>Efthimios Alepis and Constantinos Patsakis</i>	
Towards Developing Network Forensic Mechanism for Botnet Activities in the IoT Based on Machine Learning Techniques . . . . .	30
<i>Nickolaos Koroniots, Nour Moustafa, Elena Sitnikova, and Jill Slay</i>	
Performance Comparison of Distributed Pattern Matching Algorithms on Hadoop MapReduce Framework. . . . .	45
<i>C. P. Sona and Jaison Paul Mulerikkal</i>	
Robust Fingerprint Matching Based on Convolutional Neural Networks. . . . .	56
<i>Yanming Zhu, Xuefei Yin, and Jiankun Hu</i>	
A Personalized Multi-keyword Ranked Search Method Over Encrypted Cloud Data . . . . .	66
<i>Xue Tian, Peisong Shen, Tengfei Yang, Chi Chen, and Jiankun Hu</i>	
Application of Fuzzy Comprehensive Evaluation Method for Reservoir Well Logging Interpretation While Drilling . . . . .	79
<i>Zhaohua Zhou, Shi Shi, Shunan Ma, and Jing Fu</i>	
Factor Effects for Routing in a Delay-Tolerant Wireless Sensor Network for Lake Environment Monitoring . . . . .	87
<i>Rizza T. Loquias, Nestor Michael C. Tiglao, Jhoanna Rhodette I. Pedrasa, and Joel Joseph S. Marciano</i>	
Estimating Public Opinion in Social Media Content Using Aspect-Based Opinion Mining. . . . .	101
<i>Yen Hong Tran and Quang Nhat Tran</i>	
An Approach for Host-Based Intrusion Detection System Design Using Convolutional Neural Network . . . . .	116
<i>Nam Nhat Tran, Ruhul Sarker, and Jiankun Hu</i>	
A Robust Contactless Fingerprint Enhancement Algorithm . . . . .	127
<i>Xuefei Yin, Yanming Zhu, and Jiankun Hu</i>	

Designing Anomaly Detection System for Cloud Servers by Frequency Domain Features of System Call Identifiers and Machine Learning . . . . .	137
<i>Waqas Haider, Jiankun Hu, and Nour Moustafa</i>	
A Variant of BLS Signature Scheme with Tight Security Reduction . . . . .	150
<i>Tiong-Sik Ng, Syh-Yuan Tan, and Ji-Jian Chin</i>	
Quantum Authentication Scheme Based on Fingerprint-Encoded Graph States . . . . .	164
<i>Fei Li, Ying Guo, and Jiankun Hu</i>	
Cooperative Information Security/Cybersecurity Curriculum Development . . . . .	178
<i>Abdelaziz Bouras, Houssem Gasmi, and Fadi Ghemri</i>	
An Energy Saving Mechanism Based on Vacation Queuing Theory in Data Center Networks . . . . .	188
<i>Emna Baccour, Ala Gousssem, Sebti Foufou, Ridha Hamila, Zahir Tari, and Albert Y. Zomaya</i>	
Homomorphic Evaluation of Database Queries . . . . .	203
<i>Hamid Usefi and Sudharaka Palamakumbura</i>	
A Cache-Aware Congestion Control for Reliable Transport in Wireless Sensor Networks . . . . .	217
<i>Melchizedek I. Alipio and Nestor Michael C. Tiglao</i>	
A New Lightweight Mutual Authentication Protocol to Secure Real Time Tracking of Radioactive Sources . . . . .	231
<i>Mouza Ahmed Bani Shemaili, Chan Yeob Yeun, Mohamed Jamal Zemerly, Khalid Mubarak, Hyun Ku Yeun, Yoon Seok Chang, Basim Zafar, Mohammed Simsime, Yasir Salih, and Gaemyoung Lee</i>	
Fog Computing as a Critical Link Between a Central Cloud and IoT in Support of Fast Discovery of New Hydrocarbon Reservoirs . . . . .	247
<i>Andrzej M. Goscinski, Zahir Tari, Izzatdin A. Aziz, and Eidah J. Alzahrani</i>	
Performance Assessment of Cloud Migrations from Network and Application Point of View . . . . .	262
<i>Lukas Iffländer, Christopher Metter, Florian Wamser, Phuoc Tran-Gia, and Samuel Kounev</i>	
A Cloud Service Enhanced Method Supporting Context-Aware Applications . . . . .	277
<i>Zifan Liu, Qing Cai, Song Wang, Xiaolong Xu, Wanchun Dou, and Shui Yu</i>	

Application of 3D Delaunay Triangulation in Fingerprint Authentication System . . . . .	291
<i>Wencheng Yang, Guanglou Zheng, Ahmed Ibrahim, Junaid Chaudhry, Song Wang, Jiankun Hu, and Craig Valli</i>	
The Public Verifiability of Public Key Encryption with Keyword Search . . . . .	299
<i>Binrui Zhu, Jiameng Sun, Jing Qin, and Jixin Ma</i>	
Malicious Bitcoin Transaction Tracing Using Incidence Relation Clustering . . . . .	313
<i>Baokun Zheng, Liehuang Zhu, Meng Shen, Xiaojiang Du, Jing Yang, Feng Gao, Yandong Li, Chuan Zhang, Sheng Liu, and Shu Yin</i>	
Cryptanalysis of Salsa and ChaCha: Revisited . . . . .	324
<i>Kakumani K. C. Deepthi and Kunwar Singh</i>	
CloudShare: Towards a Cost-Efficient and Privacy-Preserving Alliance Cloud Using Permissioned Blockchains . . . . .	339
<i>Yandong Li, Liehuang Zhu, Meng Shen, Feng Gao, Baokun Zheng, Xiaojiang Du, Sheng Liu, and Shu Yin</i>	
Probability Risk Identification Based Intrusion Detection System for SCADA Systems . . . . .	353
<i>Thomas Marsden, Nour Moustafa, Elena Sitnikova, and Gideon Creech</i>	
Anonymizing $k$ -NN Classification on MapReduce . . . . .	364
<i>Sibghat Ullah Bazai, Julian Jang-Jaccard, and Ruili Wang</i>	
A Cancellable Ranking Based Hashing Method for Fingerprint Template Protection . . . . .	378
<i>Zhe Jin, Jung Yeon Hwang, Soohyung Kim, Sangrae Cho, Yen-Lung Lai, and Andrew Beng Jin Teoh</i>	
<b>Author Index . . . . .</b>	<b>391</b>



# Offloading of Fog Data Networks with Network Coded Cooperative D2D Communications

Ben Quinton<sup>(✉)</sup> and Neda Abutorab<sup>ID</sup>

University of New South Wales, Campbell, ACT 2612, Australia  
[quintonbj@gmail.com](mailto:quintonbj@gmail.com), [n.abutorab@unsw.edu.au](mailto:n.abutorab@unsw.edu.au)

**Abstract.** Future fog data networks are expected to be assisted by users cooperation and coding schemes. Given the finite I/O access bandwidth of the drives in the data servers and the explosive increase in the end users' demand for download of the content from the servers, in this paper, we consider the implementation of instantly decodable network coding (IDNC) in full-duplex device-to-device (D2D) enabled cooperative distributed data networks. In particular, this paper is concerned with optimizing D2D communications with efficiently coded transmissions such that we offload traffic from the expensive backhaul of network servers. Previous works implementing IDNC have not focused on a cooperative architecture, therefore a new theoretical-graph model is proposed and the optimal problem formulation is presented. However, as the optimal solution suffers from the intractability of being NP-hard, it is not suitable for real-time communications. The complexity of the problem is addressed by presenting a greedy heuristic algorithm used over the proposed graph model. The paper shows that by implementing IDNC in a full-duplex cooperative D2D network model significant reduction in the number of downloads required from the servers can be achieved, which will result in saving valuable servers' resources.

**Keywords:** Instantly decodable network coding · IoT · Full-duplex Backhaul offloading · Cooperative D2D communications  
Fog storage networks

## 1 Introduction

With the modern advancements of wireless communications, wireless networks have seen an explosion in data traffic over the past decade [6]. This rapid demand for more data is largely attributed to video and multimedia streaming, where it is expected that three-fourths of data traffic will be consumed by video [6]. To compound this further, it is expected that the next generation of wireless networks will encapsulate the new paradigm of the internet of things (IoT). This concept

moves to further integrate more and more devices into communication networks, where it is foreseen that the IoT will add a further 50 billion heterogeneous wireless devices by 2020 [6]. Consequently, this growing demand puts further pressure on data networks, where the offloading of the servers becomes an increasingly important problem.

This ever-growing demand for real-time data, where users expect to maintain their quality-of-experience (QoE) has led to much research to address the data networks backhaul problem. Multiple areas of research have shown promising methods to deal with this problem, one such option is to distribute the data closer to the users with improved redundancy [2, 7, 8, 11]. The idea of distributing resources to the edge of a network is known as “fog” networking [5]. Motivated by very high temporal correlation among the “popular” content demanded by end-users, it is expected that the proactive (i.e. without users requests) diffusing of such popular content from its storage and transmission clouds behind the backhaul, and caching it in a “fog” of low-cost storage units close to the end-users to serve the requests to download this content could largely improve the network performance and service quality. Using this approach not only the users’ requests can be immediately and efficiently addressed, but also the access to the backhaul could be significantly offloaded [10, 12].

In addition to distributing the data, with the rapid increase in the number of wireless devices, there are more and more devices in each others proximity. Such “geographically close” wireless devices form an autonomous local network over which the users can communicate and exchange files without contacting the backhaul servers. Such scenario may occur for instance when co-workers are using their tablets to share and update files stored in the cloud (e.g. Dropbox), or when users, in the subway or a mall, are interested in watching the same popular video. Under such scenario, the benefits of communicating over a local network can be utilized not only to reduce the users’ download time but also offload the backhaul of the data network (i.e., minimizing the download from it).

Furthermore, network coding (NC), initially introduced in [1], can help in offloading of the backhaul servers in the considered distributed cooperative data network scenario by maximizing the number of served users in one transmission, thus maximizing the backhaul offloading. Although NC was originally implemented at the network layer, more attractive application was found at the data link layer where there is coded combinations of files to improve throughput. Multiple areas of study have focused on various types of network coding, where this paper will focus on opportunistic network coding (ONC) [14], in particular instantly decodable network coding (IDNC) [13]. This technique has recently gained much attention due to its instant decodability (as the name suggests) by using a simple XOR operation that results in reducing the computational complexity of the decoding at the end users. It also provides a significant benefit to real-time communications, where studies in [3, 7, 13] show through a heuristic algorithm that utilizing IDNC results in shown significant performance improvements over uncoded transmissions in both centralized point-to-multipoint (PMP) and decentralized network settings.

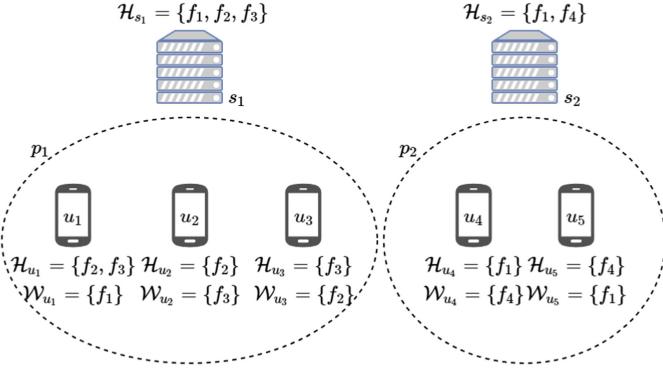
Although much work has focused on implementing IDNC in various network models to address the above problem, the studies have focused on centralized PMP and distributed architectures. Furthermore, there has currently been no work to consider implementing IDNC in a cooperative setting where there is a focus on reducing the number of downloads required from the network servers. A network coded cooperative D2D-enabled architecture is considered in this paper, as it provides an attractive solution to offload the servers in a distributed data networks. Therefore, in this paper we aim to address the following question: *How should we encode files amongst users in a cooperative D2D-enabled transmission, such that the remaining requests from the users (if any) can be delivered (using IDNC) with a minimum number of transmissions from the network servers?* Although much research has focused on implementing IDNC in a PMP setting and even a distributed architecture with multiple servers, these approaches cannot be directly applied to a full-duplex cooperative network with the current graph modelling technique, therefore there is a need to develop a new model.

To address the question above, we first need to model the problem with a new graphical representation, namely the IDNC graph with induced subgraphs. The new graph representation is developed due to limitations of the conventional graphical representation when we wish to implement full-duplex communications into the system model. With the new graph modelling of the system, the optimal solution is formulated and shown to be NP-hard and not applicable for real-time applications [9]. The paper then proposes an online greedy heuristic algorithm that employs a maximum weighted vertex search over the new graph model. Simulation results show that the proposed algorithm when employed over the new graph model in a full-duplex D2D-enabled environment significantly outperforms the conventional uncooperative IDNC approach in reducing the downloads required from the servers of distributed data networks.

In this paper, we first present the system model and mathematical notation in Sect. 2. In Sect. 3, we formulate the problem, where a motivating example is first presented then followed by a mathematical optimal solution to the problem utilizing the new graph model. As the solution is found to be NP-hard, we then present the proposed greedy heuristic scheme in Sect. 4, followed by simulation results and discussion in Sect. 5. Lastly, this paper is concluded in Sect. 6.

## 2 System Model

A distributed wireless data network model is considered in this paper and is illustrated in Fig. 1. In this model, there is a set of  $N_u$  users defined as  $\mathcal{U} = \{u_1, \dots, u_{N_u}\}$ . In the system model, the assumption is made that all users are capable of full-duplex communications. The users will request to receive one file in the current time epoch, from a library of files defined as  $\mathcal{F} = \{f_1, \dots, f_{N_f}\}$  with  $N_f$  files that are collectively stored at the servers. The servers are defined in the set  $S = \{s_1, \dots, s_{N_s}\}$  with  $N_s$  servers. All servers are assumed to have full coverage, where the users in the coverage area are denoted by  $\mathcal{U}(s_i)$  and



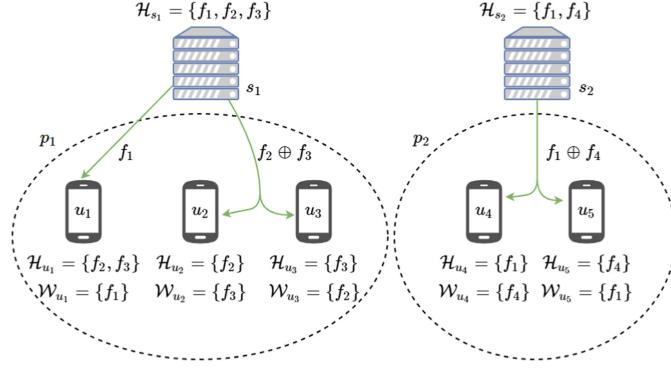
**Fig. 1.** An illustration depicting our system model for a distributed storage network, showing two servers, six users, and two proximity based wireless networks, where users can conduct cooperative D2D communications.

must satisfy  $\mathcal{U}(s_i) \cap \mathcal{U} = \mathcal{U}$ . The model shows a distributed setting where the users are in coverage of multiple servers. Also in the model, multiple proximity networks (possibly Wi-Fi or LAN) are shown. The proximity regions are defined as the proximity set  $\mathcal{P} = \{p_1, \dots, p_{N_p}\}$  with  $N_p$  proximity-enabled D2D communication networks. The proximity networks contain a subset of the users in  $\mathcal{U}$ , defined as  $\mathcal{U}(p_i)$ , that is the users in the coverage area of the proximity-enabled network  $p_i$ . It is assumed that there is no overlap of the users in each proximity set, that is, the users in each proximity network that are “geographically close” can communicate locally but not outside this network.

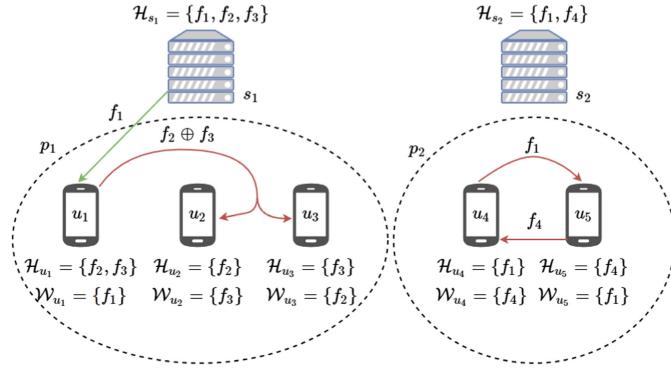
In our model, we assume the users have received some files in the initial transmission phase<sup>1</sup>. That is, a user  $u_i$  has partially downloaded some of the files from a transmitted frame which constitutes the users *Has* set  $\mathcal{H}_{u_i}$ . Furthermore, the remaining files wanted by user  $u_i$  in the frame form the user’s *Wants* set, denoted as  $\mathcal{W}_{u_i}$ . Similarly, the servers will store a subset of the files in  $\mathcal{F}$ , however the union of all files at the servers should contain the complete set of  $\mathcal{F}$  (with possible repetition). Here, a server’s *Has* set is defined as  $\mathcal{H}_{s_i}$ . It is assumed that the servers will maintain a global knowledge of the system state during the initial transmissions, that is the users will respond with positive/negative feedback depending if they receive their files successfully or not. At completion of this phase the system will move into the recovery transmission phase.

IDNC can now be utilized to exploit users’ side data to optimize the transmissions in the current time epoch. In the recovery transmission, we assume an erasure free channel, where the different users and servers will operate on orthogonal channels. It is also assumed that the servers have an unlimited capacity,

<sup>1</sup> This first phase of the transmission is known as the initial transmission phase. During the initial transmission the servers will attempt to serve all files to the users in the network. However, some users will have received only a portion of the files requested due to channel erasure.



**Fig. 2.** Conventional IDNC approach from [2], where there is a total of three downloads required for the optimal solution.



**Fig. 3.** A D2D-enabled approach showing the potential for offloading servers, where only one download is required.

such that after all cooperative D2D communications all requests remaining will be served by the server in the current time epoch. Here, the main goal is to optimize the selection of the files for network coding for the users and the servers, where priority is given to the cooperative D2D communications, such that we reduce the amount of downloads required from the servers.

### 3 Problem Formulation

#### 3.1 Motivating Example

If we consider the system model depicted in Fig. 1, it can be shown by example that by finding the optimal solution, that is, to solve our question defined earlier, there are numerous allocations of coded transmissions that can lead to different results. For way of a motivating example, we present two different allocations:

in the first solution, we assume a conventional IDNC approach without the cooperative D2D enabled transmissions. In the second solution, we show the scenario for a network with cooperative D2D enabled communications.

**Solution 1.** In this solution (depicted in Fig. 2), we employ the method used in [2], this solution utilizes a conventional IDNC approach without D2D-enabled cooperation. One possible optimal solution using this method is:

- $s_1$  transmits  $f_1$  to  $u_1$  and  $f_2 \oplus f_3$  to  $u_2$  and  $u_3$ .
- $s_2$  transmits  $f_1 \oplus f_4$  to  $u_4$  and  $u_5$ .

This scenario results in consuming three downloads from the servers.

**Solution 2.** In this solution (depicted in Fig. 3), we show a scenario where a cooperative D2D setting is incorporated. One possible optimal solution is:

- $s_1$  transmits  $f_1$  to  $u_1$ .
- $u_1$  transmits  $f_2 \oplus f_3$  to  $u_2$  and  $u_3$ .
- $u_4$  transmits  $f_1$  to  $u_5$  and  $u_5$  transmits  $f_4$  to  $u_1$ .

In the second solution, it can be seen that we only need one download from one of the servers, while no download is required from the other server. This approach shows that even in a small network setting, there is a download reduction of two thirds of the previous solution, freeing up valuable servers' resources.

Although much work has focused on implementing IDNC in various network models, the graph-theoretical modellings used in these cases are limited in a cooperative full-duplex environment. In previous approaches, the graph models incorporated assume that there is a clear differentiation between the sender and the receiver. In our setting, we remove this restriction and allow users the ability of full-duplex communications, therefore the existing IDNC graph models are not appropriate and there is a need for a new model.

### 3.2 Graph-Based Solution

To be able to formulate the optimal solution to the above problem, stated in Sect. 3, we will propose a new IDNC graph that represents coding opportunities. The IDNC graph when formulated, will represent all the possible files that can be XORed together to create a network coded transmission that can be decoded by the targeted end users. To form the model, we first define the graphs of interest in our system model as follows: Graph  $\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_{N_p}\}$  with the subgraph  $\mathcal{G}_i$  representing each discrete D2D network, as well as the graph  $\Psi = \{\Psi_1, \dots, \Psi_{N_s}\}$  that is representing all servers, where the subgraph  $\Psi_i$  represents each individual server.

To construct each of the graphs previously mentioned, we proceed as follows:

**Generate Vertex Set.** Vertices are generated from a server and user perspective under the two conditions:

- Generate a vertex set for every server  $s_i$  in  $\mathcal{S}$  that is represented in the subgraph  $\Psi_i$ , generating the vertices  $v_{ijk}, \forall s_i \in \mathcal{S}$  and  $f_k \in (\mathcal{H}_{s_i} \cap \mathcal{W}_{u_j})$ . The vertices of the subgraph are defined as  $\Psi_{i(ijk)}$ .
- Generate a vertex set for every user  $u_i \in p_n$  in  $\mathcal{U}$  that is represented in the subgraph  $\mathcal{G}_n$ , generating the vertices  $v_{ijk}, \forall u_i \in \mathcal{U}$  and  $f_k \in (\mathcal{H}_{u_i} \cap \mathcal{W}_{u_j})$  on the conditions  $u_i \neq u_j$  and both  $u_i, u_j \in p_n$ . The vertices of the subgraph are defined as  $\mathcal{G}_{n(ijk)}$ .

**Generate Coding Opportunity Edges.** In each individual subgraph in  $\mathcal{G}$  and  $\Psi$ , we connect two vertices  $v_{ijk}$  and  $v_{lmn}$  with an edge if they satisfy one of the following two conditions:

- $f_k = f_n, u_j \neq u_m$  and  $u_i = u_l$  if in  $\mathcal{G}$  (or  $s_i = s_l$  if in  $\Psi$ ), meaning the two requested files are the same, and these files are requested by two different users.
- $f_n \in \mathcal{H}_{u_j}$  and  $f_k \in \mathcal{H}_{u_m}$ , representing a potential coding opportunity, so that when  $f_n$  and  $f_k$  are XORed both users can successfully decode and retrieve their requested file.

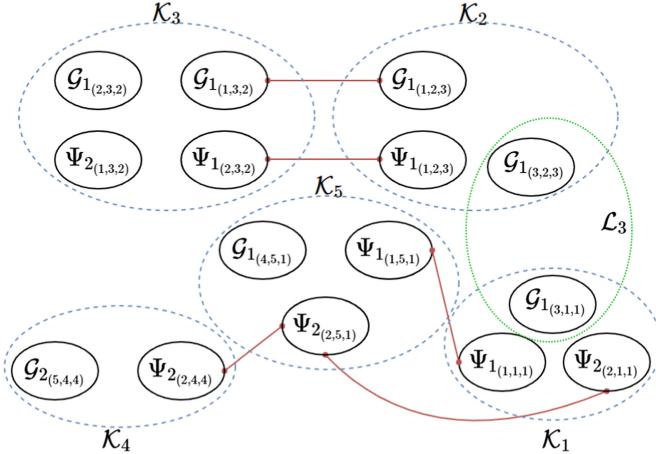
In the formulation so far we have incorporated graphs that represent coding opportunities from a user/server viewpoint. To further create a global awareness, we have to incorporate induced subgraphs (subgraphs of  $\mathcal{G}$  and  $\mathcal{H}$ ) that will represent the transmission conflicts (subgraph  $\mathcal{K}$ ) and a subgraph to ensure only one transmission per user is permitted in the current time epoch (subgraph  $\mathcal{L}$ ).

In Fig. 4, we depict the implementation of IDNC with the prescribed theoretical graph model for the example shown in Fig. 1. In our case, we show independently, firstly in subgraphs  $\mathcal{G}_i$ , the IDNC subgraphs for each individual proximity network. While in subgraphs  $\Psi_i$ , we show the potential coded transmissions in maximal cliques<sup>2</sup> for each server  $s_i$ . In the graph model shown in Fig. 4, it is clear that there is no interconnection between the graphs of  $\mathcal{G}$  and  $\Psi$  (no edges connecting vertices). Therefore, we introduce the induced subgraphs approach to allow us to represent particular conditions that need to be accounted for in our network setting. We will introduce the graph  $\mathcal{K}$ , a set of subgraphs that ensures conflict free transmissions. Additionally, we introduce the graph  $\mathcal{L}$  that contains a set of subgraphs that ensure users in a proximity network will not transmit more than once in the current time epoch.

**Generate Induced Subgraphs.** The two induced subgraphs as described are generated as follows:

---

<sup>2</sup> A clique is a sub-set of the graph, where every distinct pair of vertices in the induced subgraph are pairwise adjacent. A maximal clique is one that cannot be a subset of a larger clique [4].



**Fig. 4.** A visualization of the IDNC induced subgraph methodology proposed for system model shown in Fig. 1. The figure shows coding opportunities represented by edges, transmission conflicts represented in subgraphs  $\mathcal{K}_j$  and limitation of one transmission per user represented in subgraph  $\mathcal{L}_3$  (that is,  $u_3$  may only transmit to either  $u_2$  or  $u_1$  in the current time epoch).

- First, we define the set of induced subgraphs  $\mathcal{K} = \{\mathcal{K}_1, \dots, \mathcal{K}_D\}$  as a subset of both graphs  $\mathcal{G}$  and  $\Psi$ , where the subgraphs may contain the null-set of either  $\mathcal{G}$  or  $\Psi$  but not both. To generate the subgraph  $\mathcal{K}_j$ , each vertex  $v_{ijk}$  in both  $\mathcal{G}$  and  $\Psi$  will form a member of the subgraph  $\mathcal{K}_j$  for every vertex that has the same user  $u_j$  and file  $f_k$ .
- Similarly, we define the set of induced subgraphs  $\mathcal{L} = \{\mathcal{L}_1, \dots, \mathcal{L}_E\}$  as a subset of both graphs  $\mathcal{G}$  and  $\Psi$ . A subgraph  $\mathcal{L}_i$  is formed for any two vertices  $v_{ijk}$  and  $v_{lmn}$ , where  $u_i = u_l$  but  $u_j \neq u_m$  or  $f_k \neq f_n$ .

### 3.3 The Proposed Optimal Problem Formulation

In order to formulate the optimal solution we need to select the combination of disjoint maximal cliques from  $\mathcal{G}$  such that when these vertices are removed, thus removing the union of the associated subgraph from  $\Psi$ , we reduce the remaining maximal cliques of  $\Psi$ . That is, we wish to minimize the number of maximal cliques in  $\Psi$ , which is equivalent to minimizing the number of downloads from the servers. Therefore, we can either find an expression to minimize the maximal cliques of  $\Psi$ , or equivalently we can minimize the number of maximal independent sets<sup>3</sup> of the complementary graph of  $\Psi$ , which we refer to as  $\Psi'$ . The minimum number of maximal cliques in a graph can be found by finding the chromatic number of a complementary graph [4]. Therefore, the optimal solution can be expressed in mathematically in (1)

<sup>3</sup> An independent set is a set of vertices in a graph, no two of which are adjacent. A maximal independent set is an independent set that is not a subset of any other independent set [4].

$$\begin{aligned}
& \min_{\mathcal{I}_1, \dots, \mathcal{I}_C} \mathcal{X} \left[ \Psi' \setminus \coprod_{i=1}^C \left( \left( \mathcal{I}_i \bigcup_{j=1}^D \mathcal{K}_j \right) \cup \left( \mathcal{I}_i \bigcup_{k=1}^E \mathcal{L}_k \right) \right) \right] \\
& \text{subject to} \quad \mathcal{I}_i \subseteq \mathcal{G}, \mathcal{I}_i \cap \Psi = \emptyset \\
& \quad \exists u_l \in \mathcal{U} \text{ where } \mathcal{F}(\mathcal{I}_i) \subseteq \mathcal{H}_{u_l}
\end{aligned} \tag{1}$$

where  $\coprod$  is the disjoint set union operator and the first constraint ensures that the independent set  $\mathcal{I}_i$  is selected only from the vertices that belong to the graph  $\mathcal{G}$ . This is to ensure the selected coded file combinations that the users serve reduces the chromatic number of the servers graph (optimal solution), as the chromatic number of the remaining graph  $\Psi$  is equal to the number of downloads required from the servers. The second constraint shown in (1) ensures that for all files selected in the independent set  $\mathcal{I}_i$ , denoted by  $\mathcal{F}(\mathcal{I}_i)$ , there exists a user that posses the files and can XOR them. If this is not satisfied then the coded transmission cannot be sent and the conditions in (1) are not met.

Solving for the optimal solution that has been presented requires that we determine the chromatic number of a graph (equivalent to finding all maximal cliques). It is well known that determining the chromatic number of a graph is proven to be NP-hard [9]. This is further compounded by the fact that we not only need to calculate the chromatic number of one graph, but we need to find the selection of independent sets such that we minimize the chromatic number of the remaining subgraph. Hence, the optimal solution is not applicable for online and real-time communications. Therefore, we will propose a heuristic scheme in the following section to solve sub-optimally.

## 4 The Proposed Greedy Heuristic Algorithm

In this section, we propose a greedy heuristic approach that can be solved in real-time and efficiently reduce the number of downloads from the servers. The fall back of a greedy heuristic scheme is that it does not in fact guarantee a global optimum, although we hope that this scheme will on average, give a good approximation to it.

An attractive feature of the graph-based formulation proposed in Sect. 3.2 is that we can directly apply a maximal weighted vertex search under a greedy policy on the graph model. With the graph already established from the problem formulation, we can carry out the maximum clique listing, using a maximum weighted vertex search as follows:

Firstly, we associate a weight to each vertex in the graph  $\mathcal{G}$ , each vertex's weight is proportional to  $\delta_{ijk}$  which is the degree<sup>4</sup> of  $v_{ijk}$ . The weight is calculated in (2),

$$w_{ijk} = \sum_{v_{i'j'k'} \in \mathcal{N}(v_{ijk})} \delta_{i'j'k'} \tag{2}$$

---

<sup>4</sup> The degree of a vertex ( $\delta$ ) in a graph is equal to the number of incident edges to that vertex [4].

---

**Algorithm 1.** Algorithm for Maximum Weight Vertex Search

---

**Require:** :*Initialisation* :

- Construct Graphs  $\mathcal{G}$ ,  $\Psi$ ,  $\mathcal{K}$  and  $\mathcal{L}$
- $\mathcal{G}_s \leftarrow \{\mathcal{G}\}$
- $\Gamma \leftarrow \emptyset$

1: **repeat**

- 2:  $\forall v_{ijk} \in \mathcal{G}_s$ : Compute  $w_{ijk}$  using (2)
- 3:  $v_{ijk}^* \leftarrow \text{argmax}_{v_{ijk}} w_{ijk}, \forall v_{ijk} \in \mathcal{G}_s$
- 4: add  $v_{ijk}^*$  to  $\Gamma$
- 5:  $\mathcal{G}_s \leftarrow \mathcal{G}_s \cup v_{ijk}^*$
- 6: **until**  $\mathcal{G}_s = \emptyset$

7: **return** the clique listing  $\Gamma$ 

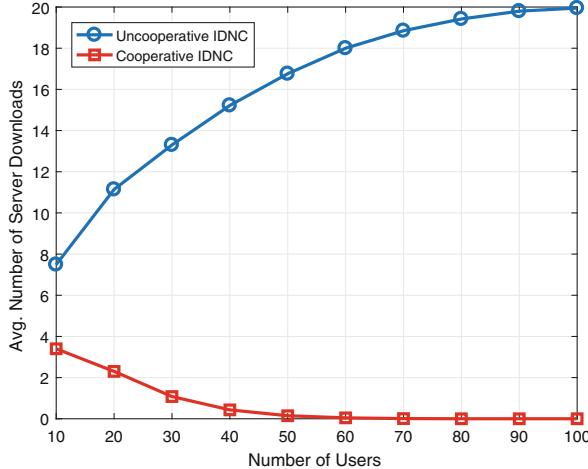
where  $\mathcal{N}(v_{ijk})$  is the set of adjacent vertices to  $v_{ijk}$ . Therefore, each vertex in graph  $\mathcal{G}$  will have a large weighting if it has a large number of adjacent vertices, which themselves have a large number of adjacent vertices. The search will then select the vertex with the largest weighting, or between those with the same largest weight with equal probability. The algorithm then removes all non-adjacent vertices to  $v_{ijk}$ , and then checks if the vertex  $v_{ijk}$  belongs to a subgraph  $\mathcal{K}_j$  or  $\mathcal{L}_i$  and will remove *all* other vertices that are a member of either subgraph.

Secondly, the algorithm will then update all weights in  $\mathcal{G}$  before selecting the next (if any) adjacent vertex in graph  $\mathcal{G}$  that forms a clique with all previously selected vertices. The algorithm then continues to iterate these steps until no more vertices can be added to the clique. Finally, once a maximal clique listing is found and removed, we iterate the whole procedure until no more vertices are left in the graph  $\mathcal{G}$ . The steps of algorithm described is summarized in Algorithm 1.

At this stage, the algorithm has removed all possible D2D cooperations available in hopes to minimize the amount of downloads from the servers. Therefore, we now need to serve the remaining vertices in graph  $\Psi$  that were not served locally from D2D cooperation. We now conduct the exact same procedure on the remaining vertices in graph  $\Psi$ , where each maximal clique represents one download from a server and continue until all vertices are removed from the graph. Once all vertices have been removed from the graph  $\Psi$  the system will have reached absorption, that is, all users will have received the file in their *Wants* sets.

## 5 Simulation Results

In this section, we present our simulation results for the proposed algorithm in a cooperative D2D setting in comparison with a uncooperative decentralized conflict free IDNC approach that was incorporated in [2]. In both cases, the aim of the approaches is to reduce the number downloads from the servers.

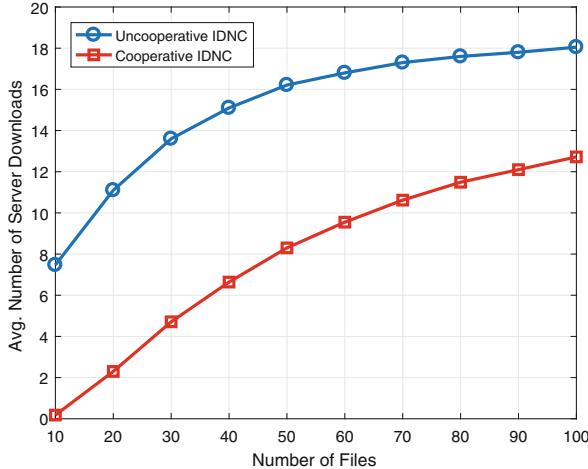


**Fig. 5.** The average number of downloads required from the servers as a function of the number of users.

In the simulations, each user is interested in receiving one file and has two files already received and stored in the *Has* set (when fixed), where the recovery downloads are to be completed in one time epoch. We assume each users' *Has* and *Wants* sets to be determined probabilistically, with uniform distribution over all files in the library. In all simulations, there are two servers available with total coverage of all users in the network, while in the cooperative model we consider a dual network where the users are split evenly between the two proximity networks  $p_1$  and  $p_2$  (similar to Fig. 1).

Firstly in Fig. 5, we show the average number of downloads required from the servers for a fixed number of files in the transmission frame of  $N_f = 20$ , as a function of the number of users  $N_u$ . The result shows that for the algorithm implemented for a cooperative D2D-enabled setting, as the number of users increase the average number of server downloads tends to monotonically decrease. Intuitively, this result is expected as more users in the network will result in a greater likelihood that the users can serve themselves independently from the servers, as the collective *Has* set of the users in the network will cover the files in the frame  $\mathcal{F}$ . Additionally, it can be seen that in comparison to a conventional uncooperative conflict-free IDNC approach, as the network size increases there is significant improvement, where we see an improvement of approximately 550% with only 20 devices in the network setting. Furthermore, approximately no downloads from the servers are required as the number of users approach 60 in this network setting, that is, 30 users in each D2D-enabled network.

Now if we consider fixing the number of users to 20, while varying the amount of files per transmission frame, we can see the results in Fig. 6 for cooperative versus uncooperative IDNC transmission schemes. In both cases, it can be seen as the number of files increase, both schemes show a similar increase on the



**Fig. 6.** The average number of downloads required from the servers as a function of the number of files.

number of downloads required from the servers. Although the two schemes tend to converge if we consider an asymptotic limit, the cooperative scheme still shows reasonable improvement of approximately 50% for up to 100 files. Again, this result is expected as increasing the number of files in a frame reduces the potential to leverage a coded transmission. Additionally, as the number of files increase the likelihood of a users ability to diffuse the wanted packets is diminished. Nevertheless, the cooperative approach still shows significant ability to reduce the number of downloads required from the network servers.

## 6 Conclusion

In this paper, we investigated the problem of offloading the expensive backhaul of data network servers through a network coded cooperative D2D network model. The problem was formulated using the IDNC induced subgraph model, where the optimal solution requires finding maximal cliques of multiple graphs. In the problem formulation it is found that an optimal solution is intractable and not solvable in real-time, therefore a greedy heuristic algorithm is employed using a maximum weighted vertex search approach. The paper utilizes the proposed subgraph model again in the heuristic approach, where the simulation results showed a significant improvement over the conventional method that incorporates IDNC in a distributed fashion without D2D enabled cooperation.

## References

1. Ahlswede, R., Cai, N., Li, S.-Y.R., Yeung, R.W.: Network information flow. *IEEE Trans. Inf. Theory* **46**(4), 1204–1216 (2000)
2. Al-Habob, A.A., Sorour, S., Abutorab, N., Sadeghi, P.: Conflict free network coding for distributed storage networks. In: 2015 IEEE International Conference on Communications (ICC), pp. 5517–5522. IEEE (2015)
3. Baran, P.: On distributed communications networks. *IEEE Trans. Commun. Syst.* **12**(1), 1–9 (1964)
4. Bondy, J.A., Murty, U.S.R.: Graph Theory with Applications, vol. 290. Macmillan, London (1976)
5. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of 1st Edition of the MCC Workshop on Mobile Cloud Computing, pp. 13–16. ACM (2012)
6. Cisco: Cisco visual networking index: global mobile data traffic forecast update. Technical report, February 2016
7. Dimakis, A.G., Godfrey, P.B., Wu, Y., Wainwright, M.J., Ramchandran, K.: Network coding for distributed storage systems. *IEEE Trans. Inf. Theory* **56**(9), 4539–4551 (2010)
8. Dimakis, A.G., Ramachandran, K., Wu, Y., Suh, C.: A survey on network codes for distributed storage. *Proc. IEEE* **99**(3), 476–489 (2011)
9. Edwards, C.S., Elphick, C.H.: Lower bounds for the clique and the chromatic numbers of a graph. *Discret. Appl. Math.* **5**(1), 51–64 (1983)
10. Golrezaei, N., Molisch, A., Dimakis, A.G., Caire, G.: Femtocaching and device-to-device collaboration: a new architecture for wireless video distribution. *IEEE Commun. Mag.* **51**(4), 142–149 (2013)
11. Papailiopoulos, D.S., Luo, J., Dimakis, A.G., Huang, C., Li, J.: Simple regenerating codes: network coding for cloud storage. In: 2012 Proceedings of IEEE INFOCOM, pp. 2801–2805. IEEE (2012)
12. Shammugam, K., Golrezaei, N., Dimakis, A.G., Molisch, A., Caire, G.: FemtoCaching: wireless content delivery through distributed caching helpers. *IEEE Trans. Inf. Theory* **59**(12), 8402–8413 (2013)
13. Sorour, S., Valaee, S.: On minimizing broadcast completion delay for instantly decodable network coding. In: 2010 IEEE International Conference on Communications (ICC), pp. 1–5. IEEE (2010)
14. Sorour, S., Valaee, S.: An adaptive network coded retransmission scheme for single-hop wireless multicast broadcast services. *IEEE/ACM Trans. Netw. (TON)* **19**(3), 869–878 (2011)



# Persistent vs Service IDs in Android: Session Fingerprinting from Apps

Efthimios Alepis<sup>(✉)</sup> and Constantinos Patsakis

Department of Informatics, University of Piraeus,  
80, Karaoli & Dimitriou, 18534 Piraeus, Greece  
[talepis@unipi.gr](mailto:talepis@unipi.gr)

**Abstract.** Android has conquered the mobile market, reaching a market share above 85%. The post Lollipop versions have introduced radical changes in the platform, significantly improving the provided security and privacy of the users. Nonetheless, the platform offers several features that can be exploited to fingerprint users. Of specific interest are the fingerprinting capabilities which do not request any dangerous permission from the user, therefore they can be silently shipped with any application without the user being able to trace them, let alone blocking them. Having Android AOSP as our baseline we discuss various such methods and their applicability.

## 1 Introduction

Mobile devices, especially smartphones have become an indispensable part of our daily lives, as a big part of our communications and daily activities is processed and monitored by them. One of the main reasons for their wide adoption is that they have a plethora of embedded sensors that allow them to understand their context and adapt accordingly. For instance, through luminosity and proximity sensors as well as accelerometers, mobile phones may adapt the UI to fit better to user expectations. Moreover, thanks to GPS, mobile devices are location aware enabling them to render content according to the spacial restrictions significantly improving the user recommendations.

Data mining and data profiling can be used in order to collect valuable information about a particular user or group of users, in order to generate a profile [12], which can be further used by companies to gain profit. As stated in [14], this kind of information, namely user profiling, is valuable also for advertisers who want to target ads to their users and in return, advertisers may pay more money to their hosting applications' developers. Building user profiles requires, as the authors state, sensitive privileges in terms of permissions, such as Internet access, location, or even retrieving installed applications in a user's device [14]. To this end, we may infer that collecting and successfully fusing user data from more than one service can create even better and more complete user profiles, which will consequently translate in higher monetization. Looking back in 2009, it was quite clear that:

“Once an individual has been assigned a unique index number, it is possible to accurately retrieve data across numerous databases and build a picture of that individual’s life that was not authorised in the original valid consent for data collection” [19].

The above has been realised by tech giants. For instance quoting a statement from Google’s current privacy policy [10]:

“We may combine personal information from one service with information, including personal information, from other Google services - for example to make it easier to share things with people you know. Depending on your account settings, your activity on other sites and apps may be associated with your personal information in order to improve Google’s services and the ads delivered by Google”

In order to “enable” data fusion from different sources and services, one could argue that unique identifiers should be either implicitly or explicitly present. In particular, a value describing a quantity of some valuable variable might be useless if it is not accompanied by a unique identifier that would allow us to track its source. Contrariwise, identifiers coming from different services that are matched, may act as a “bridge” between these services to combine their corresponding datasets and integrate them.

During the last decade relevant surveys have revealed that the majority of both iOS and Android apps were transmitting the phone’s unique ID and the user’s location to advertisers. These findings are confirmed by “The Haystack Project” [11] which revealed that nearly 70% of all Android apps leak personal data to third-party services such as analytics services and ad networks [20].

All this wealth of information apart from the benign usage for the user benefit has been a constant target by companies who wish to monetize it, mainly through targeted advertisement. The recent advances in big data and data mining have enabled the extraction of information from theoretically diverse data, leading to the revealing of a lot of sensitive data through data fusion. To this end, many fingerprinting techniques have been introduced in order to link data flows to specific individuals. Apparently, since Android is currently the prevailing mobile platform, most companies are targeting it with their apps, under the freemium model, harvesting user data to monetize them. There is even a common saying in the privacy community suggesting that “If you’re not paying for the product, you are the product”. To this end, If users are not paying for an app, they are usually selling their profiles (with or without their knowledge/consent) to an ad network, which will use their unique identifiers to track and target them.

In view of the above and targeting at improving the OSes privacy, the new coming Android O, makes a number of privacy-related changes to the platform, many of which are related to how unique identifiers are handled by the system [9], and in particular aiming to help provide user control over the use of identifiers [2]. One of the most important improvements concern “limiting the use of device-scoped identifiers that are not resettable”.

Clearly, during app environment of Android hardware identifiers can greatly facilitate companies' attempt to deanonymise users. Therefore, Google has been gradually introducing specific measures to restrict them. In fact, Google decided to introduce further restrictions in one more identifier; not hardware based, namely `Android_ID`. While this attempt might seem noble, in this work we show that these restrictions do not actually serve the purpose, while apps may be deprived of many persistent identifiers, ephemeral IDs can actually serve their purposes in the attempt to deanonymize their users and fuse their data.

### 1.1 Main Contributions

The main contribution of this work is to study user fingerprinting from mobile apps, which correspondingly and as already discussed can lead to user profiling. In this regard we assume that apps and software services which profile users, want to correlate the information that each one of them has collected to fine tune their profiles. Conceptually, in order to provide proof for our claims, we explore all the available communication channels that mobile apps could utilize in Android AOSP in order to identify that specific profiles are installed in the same device. Furthermore, we suggest that the existing underlying mechanism is able to function without using unique hardware identifiers, nor dangerous permissions which could alert the user, or demand further user interaction. While many of the communication mechanisms are apparent, e.g. inter-process communication, there is a wide misconception that the upcoming changes in Android O will eradicate many such issues. Therefore, initially we analyse each possible communication channel and ways it can be used to transfer the needed information. Moreover, by providing statistical evidence we discuss when these changes are expected to be noticed by the average user. Finally, despite the touted changes in `Android_ID`, we detail new methods that can provide permanent cross-app IDs that can be collected even if an app is uninstalled.

### 1.2 Organisation of This Work

The rest of this work is organized as follows. In the next section we present the related work. Section 3 provides the problem setting and our basic assumptions. Then, Sect. 4 presents all the available communication channels. In Sect. 5 illustrates possible temporary and ephemeral identifiers that can be used to link users between applications using Android AOSP as our reference point. Finally, the article concludes with some discussion about our research findings and providing statistics regarding the adoption timeline of the expected anonymization mechanisms of Android O.

## 2 Related Work

Unique identifiers have been used for a long time and facilitate many tasks in modern database systems as they allow us to perform record linkage between

different entities and extract the necessary information and thus knowledge from the corresponding database tables. The most typical example of a unique identifier is the Social Security Number, which allows us to distinguish two people from each other. However, in the digital era, unique identifiers can be considered hardware identifiers like the MAC address of the network card, or a set of properties such as browser fingerprints which consist among others of the browser version, OS, fonts, and browser plugins.

In the Android ecosystem there is a plethora of unique identifiers which have so far been extensively exploited by advertisement companies to track users and their interests as ad libraries have become more and more greedy and rogue [3,18] while apps may deliberately leak information to the ads [4,21] harnessing arbitrary amounts of users' sensitive information directly or indirectly [5]. A key role in this procedure is the use of unique identifiers [17]. Acknowledging this situation, Google initially introduced some recommendation guidelines for the proper use of unique identifiers in Android [1]. Then, Google gradually started requesting more permissions from the apps to allow them access to these identifiers. For instance, a typical unique identifier for mobile phones are IMEI and IMSI, however, after Marshmallow, the user has to grant the dangerous `READ_PHONE_STATE` permission to an app to access them. While many users may ignore app permissions [8], for many others it works as an obstacle, forcing many companies to comply with the rule.

Despite the ads, apps may collaborate in order to perform malicious acts which independently would not be allowed to perform. Orthacker et al. [15] study this problem from the aspect of permissions. In this regard, the malicious apps which are installed in the victim's device may result in "possessing" and correspondingly using dangerous permissions that other normal apps do not. The concept is that the user would not allow camera and microphone permission to a single app. However, since the permissions are requested by two apps which are seemingly independent, the permissions are "spread" so the user grants them, yet an adversary controls both of them getting access to the desired resource. Contrary to Orthacker et al. we do not aim to resources, but access to information that the user would not share to one specific app to prevent his profiling.

In Nougat, the current stable version of Android, Google prohibited unprivileged access to even more hardware identifiers, such as the MAC address of the WiFi card, by restricting access to `/proc`. While the latter measure creates many issues with applications targeting towards security and privacy services as Google has not provided any permission so far to access this information, undoubtedly, it leaves little space to adversaries to exploit.

### 3 Temporary and Ephemeral Identifiers

#### 3.1 Problem Setting

While the aforementioned issues have led to the introduction of many changes to Android, improving the security and privacy of the OS, in terms of user fingerprinting, from the side of apps, we argue that little has been achieved.

Certainly, direct access and/or unprivileged access to hardware identifiers has been removed, therefore, permanent or long term identifiers are not going to be available in the coming version of Android, nonetheless, this is not what the advertisement companies are actually trying to do. Undoubtedly, such access facilitates the correlation process, nonetheless, it is a great misconception to consider that a unique device identifier from a device is all that two apps need to correlate user information. More specifically, owning a unique identifier, however not being able to communicate it to others cannot be considered a threat. Similarly, having access to a communication channel, yet failing to uniquely identify the transmitted data results in data loss. Randomly generated identifiers, locally stored in apps, as it will be shown, offer a solution, however also suffer from lack of persistency. In this work we present methods which bypass these obstacles and result in identifying users and also allow communication of this information to other parties.

### 3.2 Basic Assumptions and Desiderata

In what follows we assume that the user has installed at least two applications in his device. In the same sense, this approach can be generalized in software services that communicate and/or handle a number of mobile apps. Our reference is Android AOSP as it provides all the baseline security and privacy methods therefore all derivative versions may have glitches which are vendor specific, perhaps apart of CopperheadOS<sup>1</sup> which is a hardened version of Android. Moreover, in order to highlight the magnitude of the presented privacy issue, we further assume that the two aforementioned applications did not request any permission from the user during installation, nor during runtime. We also assume that these apps do not belong to the same developer, nonetheless, the developers have decided to cooperate in exchanging user data to create a more fine-grained profile of their users. Finally, we assume that even in the cases where the user has authenticated himself to each app, for each of them he uses completely different credentials e.g. the username is different in both apps. One can easily deduce that by “relaxing” these assumptions, our work becomes much easier.

Apparently, one way to achieve their goal, the app developers only need to determine that the two apps are running in the same device and exchange the corresponding IDs. Inarguably, the two apps do not “care” whether the user has bought a new device and installed both of them there, since their goal is to extend the user profile that each one of them has created by fusing all the available information. This profile spans throughout a session, therefore, their goal is to anonymize a session, regardless of its span. If they manage to exchange the user IDs, then the developers may use them to request the needed information from each other. Note that due to the nature of Android Package Manager class, all apps are aware of which apps are installed in the system without requesting any permission. Therefore, the challenge lies in the exchange of the user IDs through a communication channel.

---

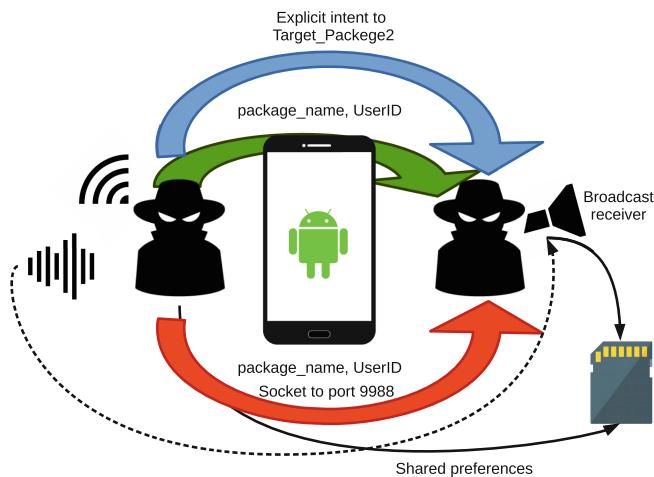
<sup>1</sup> <https://copperhead.co/android/>.

For the scope of clarity, in what follows we omit the use of e.g. encryption and digital signatures to hide the content of exchanged messages or to verify their source and authenticity. Moreover, we study each method independently considering how one could exchange the needed information, even if the previous ones did not exist.

Finally, it is apparent that even if the user does not authenticate to the app, hence there is no directly linked user ID, the app may create a random ID and use it as the session ID. Since this is linked to the session and can be stored by the app in its storage space, this ID can also serve as the user ID for the lifespan of the app. However, as it will be discussed in the next sections, a random ID can also serve in the cases of local communication between apps, while for remote communication further measures should be taken.

## 4 Exchanging Unique Identifiers

In the following paragraphs, we discuss methods that allow apps to correlate user information without using unique hardware identifiers. More importantly, all the methods which are described do not require any permission; let alone dangerous ones, as they depend on inherent Android mechanisms and structures, so no user interaction is required. The basic overview of the proposed methods is illustrated in Fig. 1, while code snippets that provide these functionalities can be found in the Appendix and illustrate not only how easy they can be applied, but also that many of them could be realised through reflection to avoid static code analysis.



**Fig. 1.** Proposed fingerprinting methods

## 4.1 Sockets

In Android each application corresponds to a different user and is executed in an isolated VM. Therefore, apps cannot directly access the data of each other. The same applies for their temporary files, which actually are stored in the protected installation directory of each app under `/data`. Since they cannot directly share a file using the filesystem, an obvious way to exchange some information between two apps is by opening a socket. Socket programming is one of the most fundamental primitives in every Unix-like system, and despite many changes, Android is still one of them. While sockets are very common in Android, many security issues have been raised [6, 13]. For our scenario the case is rather straight forward, as one of the apps needs to open a socket in a predefined port and await for connections which will transmit a message of the form:  $(\text{package\_name}, \text{ID}_1)$ . Clearly, to make these methods more stealth and secure, port knocking could also be considered along with encryption to counter man-in-the-middle attacks. The response will contain the ID of the other app, allowing both developers to request the desired data from each other.

## 4.2 Android IPC

As already discussed, Android apps belong to different users. Moreover, apps and system services run in separate processes. This restriction actually improves the security, stability, and memory management of the system. For instance, since each app runs as a different process of another user, should the system regard it as unnecessary or functioning improperly, it can easily “kill” it without jeopardizing further dependencies, allocating immediately the freed resources to the system.

To facilitate Inter-process communication (IPC), Android uses the binder framework (Binder) which exposes simple to use APIs. Due to its critical role, the security of Binder has been studied, revealing major security issues [16]. Some of the most generic Android mechanisms, such as Intents, Services, Content Providers, Messenger, and also common system services like Telephony and Notifications, utilize IPC infrastructure provided by the Binder framework.

One of the most profound ways to use Binder for our work is intents. More precisely, since we assume that each one of the two apps knows that the other is installed, the use of explicit intents is apparent. A similar approach can also be implemented through the use of Broadcast Receivers who are associated with intents due to the use of the Binder. Broadcast Receivers allow applications to register for system or application events. Therefore, once a registered event happens, the corresponding receivers are notified and a result can be transmitted to them. In this regard, the two cooperating apps may agree upon an app event, so that they can register to each other and exchange the required information.

Bound services provide another straightforward solution, as they represent a “server” component in a client-server interface. Bound services allow components, such as activities of other apps to bind to a service, send requests, receive responses, and perform IPC. A typical bound service may be utilized in order to

serve another application component, running in the background. In the same sense, the messenger interface provides another well-defined IPC infrastructure that enables mutual authentication of the endpoints, if required.

Finally, a more Linux-based approach would be to use shared memory. This operation however is quite similar to the Binder-based approach, that provides a “wrapper” for the more complicated remote procedure calls (RPC) mechanisms.

### 4.3 Shared Preferences

Modern applications are not static and in order to adapt according to the user preferences, they need a registry to keep track of them and cater for future changes. In Android, to keep the preferences of each app isolated from the others, this registry is held in the private folder of each app in the form of an XML file. More precisely, in a file named `/data/data/package_name/shared_prefs/filename.xml`. These files however can be tagged as `MODE_WORLD_READABLE` and/or `MODE_WORLD_WRITEABLE` allowing other apps to read and write data, if they are aware of where they are stored. Apparently, the two cooperating apps can use this mechanism in conjunction with encryption to exchange the corresponding user IDs. Clearly, the exact same mechanism could be used to exchange information with a temporary file stored in the SD card, however, for the latter a dangerous permission is required.

### 4.4 Clipboard

Clipboard is one of the most widely used features in GUIs as it enables users to seamlessly copy information from one app to another. This is rather important in Android due to the size constraints of the device it usually operates, where typing is not as easy as in common desktop computers. Clearly, adding some information in a public readable and writable channel such as the clipboard, implies several risks which can be easily exploited [7]. In the case of Android, apps do not have to request any permission to access the clipboard, but additionally they can subscribe to receive clipboard change events allowing them collect the shared information, as well as append their data. Clearly, using the proper format and encryption, two cooperating apps can easily exchange the needed information using the clipboard.

### 4.5 Internet

Utilizing the Internet for communication is probably one of the most obvious solutions. Applications having harvested user data, aim foremost to transmit them to a remote service for further processing. To this end, the “Internet” permission is required, however, as a “normal” permission (from Marshmallow and above), this requires no specific user action, nor can it be withdrawn. Interestingly, since the last Android versions, this permission is found as the most

“used” one in all the available applications. Nevertheless, having a number of apps accessing the Internet does not necessarily imply that these apps are able to exchange information about a specific user. The main reason for this is because the aforementioned “techniques” reside inside a mobile device, they represent local communication channels. On the contrary, Internet access is not “local” and involves a chaotic number of possible endpoint combinations. Even in the case where all apps point to a specific web server, there is always the challenge of determining which of the available apps reside in the same device. For this reason, in order for two or more apps to establish a communication channel between them in order to exchange user specific data, unique device or user identifiers should be present.

#### 4.6 Sensors

While dangerous permissions require user’s consent, normal permissions are automatically granted and cannot be revoked. Theoretically, these permissions do not imply any security and privacy threat for the user, nonetheless, they can be used to deduce other sensitive information. For instance, the acceleration sensor does not request any permission to be used, however, it can be used as a covert channel to slowly receive information, if combined with the corresponding vibration pattern. In our use case, we assume that a trusted third party issues a request to all apps named  $pkg_1$  to wait to receive a vibrating signal which matches a specific pattern. The pattern is triggered by  $pkg_2$  which turns on vibration (through a normal permission) and encodes in the form of e.g. Morse code the aforementioned pattern. Should one installation of  $pkg_1$  detect this pattern, then the apps can exchange the corresponding user ID. Similarly, instead of the vibration/accelerometer pair, one could use light/luminosity combination or try to correlate the sensed information at a given timeframe. While this task could be achieved, the clustering effort implies a lot of communication from each device making the method less practical. Alternatively, after exchanging a unique identifier through sensors, the cooperating apps could utilize the Internet communication, as already discussed, in order to communicate.

### 5 Identifiers

This problem of accurately identifying whether two or more mobile applications reside in the same device may be resolved by accurately matching the available identifiers of the apps. However, as already discussed, it becomes apparent that these identifiers cannot be randomly generated by the apps, since there are cases where they would never match (e.g. two random IDs from two different apps). The required identifiers should become available by a more “generic” entity, that is the user in question. To this end, the following subsections describe possible ways of deanonymizing users, without using hardware identifiers.

## 5.1 Procs Information

Exploiting the concept of a trusted third party which orchestrates the exchange of the collected information, apps can use other “public” information which is shared in Android AOSP. A typical example is the uptime which indicates how many milliseconds the device is running since last boot. This information can be retrieved by the corresponding API call, or through the `/proc/uptime` file. While the apps may not collect this information simultaneously, so the clusters may contain many possible pairs, this can be overpassed by reading both the uptime and the current-time timestamps and making the required subtraction. This operation can also be easily improved by acquiring additional public information such as battery status. Interestingly, other unique, session specific, identifiers, can be found in `procfs` many of which may even be vendor specific. An attempt to map the variations of Android can be found in Android Census<sup>2</sup>. Some of the most profound and wide spread world readable files that can be used for deanonymization in `/proc` are listed below. In each case we provide an overview of the contents and how it can be used to allow to apps that they simultaneously operate in the same device, creating a session ID.

- `boot_stat`: Contains statistics about the boot process. Due to the randomization of the process IDs and the randomness of running times of each process, it is highly impossible for two devices to have the same statistics, even if the vendor is the same.
- `diskstats`: This file, as the name suggests, contains information about the disc usage. Again, two devices are not expected to have the same statistics. However, since these statistics are subject to time constraints, if the two cooperating applications do not take the snapshots simultaneously, minor changes may appear.
- `interrupts`: The file contains information about the interrupts in use and how many times the processor has been interrupted. Small variations of the contents may appear from the timing of the snapshots.
- `meminfo`: Similar to `diskstats`, but for memory usage.
- `pagetypeinfo`: Keeps track of free memory distributions in terms of page order. As in `diskstats`, minor variations may appear due to snapshot timing.
- `stat`: Here several statistics about kernel activity are recorded. Similar to the case of `diskstats`.
- `usblog`: This file keeps track of USB usage and can be used for fingerprinting due to the stored timestamps.
- `vmstat`: This file keeps virtual memory statistics from the kernel, hence minor variations are expected.
- `zoneinfo`: This file provides details about the memory management of each zone, so minor variations are expected due to timing of the snapshots.

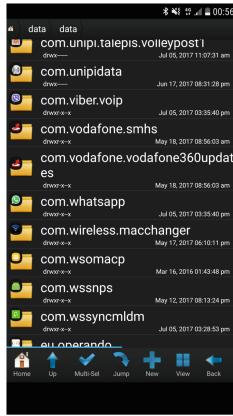
---

<sup>2</sup> <https://census.tsyrklevich.net/>.

## 5.2 Application Metadata

As already discussed, the Android's Package Manager class is capable of reporting all the installed apps to anyone requesting this information without requesting any permission. Moreover, apps can subscribe to the system event of app installation to be notified when other apps are installed and update the list of installed apps accordingly. Theoretically, this information can be used as a device fingerprint since this information is expected to differentiate two users. More interestingly though, while the `/data/pkg_name` is by default private and cannot be accessed by other apps, the exact creation date of each folder, as well as the folder's size can be retrieved for any app folder providing the necessary identifying information. Figure 2 illustrates the creation dates of a number of directories in Android regarding application installations.

Moreover, even if the set comprising of all the applications' metadata changes during time (e.g. new app installations and/or app uninstallations), its subsets can be still used as unique identifiers. A number of installed applications within a mobile device, accompanied with their installation date and their folder size can be considered a unique identifier. Furthermore, considering the fact that users have a "tendency" to use specific apps in each device they own, and even more, they most probably install the apps they have already purchased from importing their profile in Google Play Services in each new mobile device, we may safely assume that the apps' package names, especially the paid ones, within a mobile device may further uniquely identify users.



**Fig. 2.** Installed packages and the creation date of the folders

## 6 Discussion

The upcoming version of Android, dubbed "O" at the moment of writing, introduces a significant change for one of the identifiers that Google was promoting,

namely `Android.ID`, a 64-bit value. Up to Nougat, `Android.ID` was scoped per user, but since most devices had one user, this ID was actually a unique identifier for the device. Notably, this identifier was generated on first boot, or user creation, and was expected to change only once the user wiped his device. However, in “O” this changes radically, as `Android.ID` is now scoped per-app. More precisely, the `Android.ID` is computed based on the package name, the signature, the user, and the device, theoretically deterring apps from correlating user information.

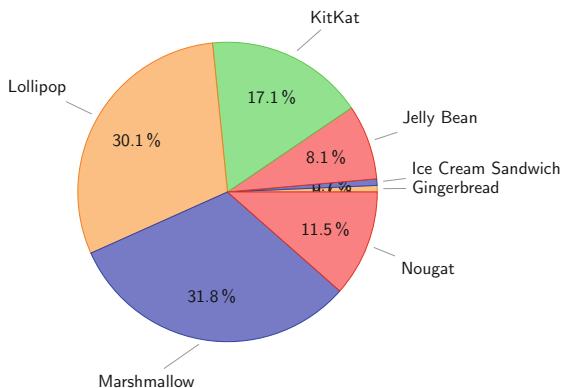
However, in Android “O” Google decided to add even more restrictions than Nougat. Therefore, the `android.os.Build.SERIAL` which returns the hardware serial also requires the dangerous `PHONE` permission. Moreover, the `ro.runtime.firstboot` property is no longer available as well as other identifiers such as `persist.service.bdroid.bdaddr` and `Settings.Secure.bluetooth_address` related to the Bluetooth MAC address and a camera hardware identifier for some HTC devices `htc.camera.sensor.front_SN`. Finally apps can get the MAC address of the WiFi card only if they are granted the `LOCAL_MAC_ADDRESS` permission.

The above changes significantly remove many capabilities of apps in collecting device specific unique identifiers. Table 1 illustrates these changes, however, many forms of transferring the needed data, especially the ones proposed by the authors, still remain.

**Table 1.** Applicability of fingerprinting methods.

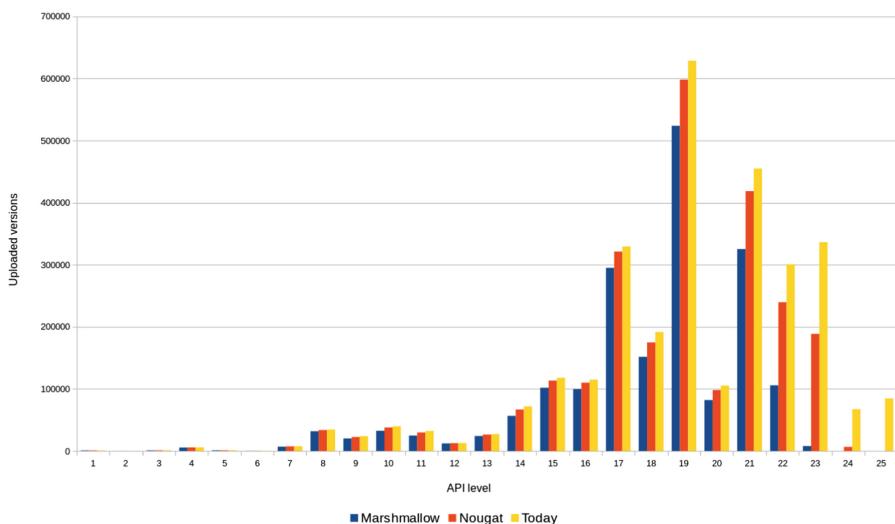
	Marshmallow	Nougat	O
Sockets	X	X	X
Binder-based methods	X	X	X
Broadcast receivers	X	X	X
Clipboard	X	X	X
Shared preferences	X	X	
Android ID	X	X	
Procfs information	X	X	
Sensors	X	X	X
Application metadata	X	X	X

Since many of the aforementioned changes will be introduced to Android O, one major question is when the user is expected to experience them. To answer this question one needs to consider the Android diversity. Apart from the different vendor flavors that Android comes, the system is highly fragmented, see



**Fig. 3.** Android versions market share as of time writing Source: <https://developer.android.com/about/dashboards/index.html>.

Fig. 3. To determine how fast developers redesign their apps to provide features that new versions of Android offer, we used data from Tacyt<sup>3</sup> a platform from ElevenPaths which downloads and analyses each application's versions from Google Play and others. The reported results in Fig. 4 are per version and illustrate which API level each version is targeting. In blue and red the reference



**Fig. 4.** Target API level per release date (Marshmallow, Nougat) and till today.

<sup>3</sup> <https://tacyt.elevenpaths.com>.

dates are the release dates of Marshmallow and Nougat respectively, while yellow represents the versions to date. While the retrieved information refers to versions and not apps, the reader can easily monitor the resulting trends. It can be observed that even after 16 months since the introduction of Nougat; the current stable version, not only few users have switched to it, but even fewer apps have integrated these features. While the most targeted API is 19, new developers' targeting is towards Marshmallow (API level 23) which dates back to October 5, 2015. Following this trend, one may speculate that the transition of apps to Android O is expected to take about two years, therefore the mechanisms discussed in this work not only currently affect millions of users, but are expected to stand for the years to come.

Finally, it is worthy to note that from the aforementioned fingerprinting capabilities, app metadata still constitute a novel unique identifier which remains even if an app is uninstalled. In this regard, app metadata can serve not only as an alternative unique ID to Android ID, but also as an alternative to Advertiser ID, working on all Android versions up to O. In fact, while Advertiser ID is user-resettable, app metadata are persistent and can only be erased after factory reset. Therefore, Table 2 summarizes the persistence of each identifier, whether it can be reset, and whether a dangerous permission is required.

**Table 2.** Unique identifiers and their persistence.

Unique identifier	Post O era	Persistent	User-resettable	Dangerous Permission
Android ID	X			X
MAC	X	X		X
Advertising ID	X	X	X	?
Build SERIAL	X	X		X
App metadata	X	X		
IMEI	X	X		X
IMSI	X	X		X
IP addresses	X			X
GSF android.ID	X	X		X
Contact profile	X	X	X	X

**Acknowledgments.** This work was supported by the European Commission under the Horizon 2020 Programme (H2020), as part of the *OPERANDO* project (Grant Agreement no. 653704). The authors would like to thank *ElevenPaths* for their valuable feedback and granting them access to Tacyt.

## Appendix

### Sample Code

Category	Sender	Receiver	Required Permissions
Explicit Intents	<pre>intent.putExtra("msg", "Some Data"); startActivity(intent);</pre>	<pre>String s = getIntent().getStringExtra("msg");</pre>	None
Intents Returning Results	<pre>intent.putExtra("package", "Package1"); intent.putExtra("ID", "123456789"); startActivityForResult(intent, request_code);"</pre>	<pre>intent.putExtra("msgResult", "Some data"); setResult(RESULT_OK, intent); finish();"</pre>	No
Local Sockets	<pre>ls.connect(new LocalSocketAddress(     SOCKET_ADDRESS)); String msg = "Some Data"; ls.getOutputStream().write(msg.getBytes()); ls.getOutputStream().close();"</pre>	<pre>LocalSocket ls = server.accept(); InputStream input = ls.getInputStream(); int readbytes = input.read();"</pre>	No
Remote Sockets	<pre>Socket socket = serverSocket.accept(); OutputStream outputStream; outputStream = socket.getOutputStream(); String msg = "Some Data"; PrintStream ps = new PrintStream(outputStream); ps.print(msg); ps.close();"</pre>	<pre>InputStream input = socket.getInputStream(); int readBytes = input.read(); socket.close();"</pre>	Internet
Bound Services	<pre>bindService(intent, mConnection, Context.     BIND_AUTO_CREATE);"</pre>	<pre>public IBinder onBind(Intent intent) {return     mBinder;}"</pre>	No
Broadcast Receivers	<pre>intent.setAction(SOME_ACTION); intent.putExtra("msg", "Some Data"); sendBroadcast(intent);"</pre>	<pre>String s = arg1.getExtras().getString("msg"); }"</pre>	No
App Local Storage	<pre>Editor edit = prefs.edit(); edit.putString("msg", "Some Data"); edit.commit();"</pre>	<pre>SharedPreferences prefs = context.     getSharedPreferences("SP", Context.         MODE_WORLD_READABLE); String s= prefs.getString("msg", "No Data found ");"</pre>	No
Device Storage	<pre>String msg = "Some Data"; fos.write(msg.getBytes()); fos.close();"</pre>	<pre>DataInputStream dis = new DataInputStream(fis); BufferedReader br = new BufferedReader(new     InputStreamReader(dis)); String s = br.readLine(); in.close();"</pre>	Read/Write Storage
Clipboard	<pre>clipboardManager = (ClipboardManager)     getSystemService(CLIPBOARD_SERVICE); ClipData clipData; clipData = ClipData.newPlainText("msg", "Some     Data"); clipboardManager.setPrimaryClip(clipData);"</pre>	<pre>clipboardManager = (ClipboardManager)     getSystemService(CLIPBOARD_SERVICE); ClipData clipData = clipboardManager.         getPrimaryClip(); ClipData.Item item = clipData.getItemAt(0); String s = item.getText().toString();"</pre>	No

## References

1. Android Developers: Best practices for unique identifiers. <https://developer.android.com/training/articles/user-data-ids.html>. Accessed 5 July 2017
2. Android Developers Blog: Changes to device identifiers in Android O. <https://android-developers.googleblog.com/2017/04/changes-to-device-identifiers-in.html>. Accessed 24 July 2017
3. Book, T., Pridgen, A., Wallach, D.S.: Longitudinal analysis of android ad library permissions. arXiv preprint [arXiv:1303.0857](https://arxiv.org/abs/1303.0857) (2013)
4. Book, T., Wallach, D.S.: A case of collusion: a study of the interface between ad libraries and their apps. In: Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices, pp. 79–86. ACM (2013)

5. Demetriou, S., Merrill, W., Yang, W., Zhang, A., Gunter, C.A.: Free for all! assessing user data exposure to advertising libraries on android. In: NDSS (2016)
6. Fahl, S., Harbach, M., Muders, T., Baumgärtner, L., Freisleben, B., Smith, M.: Why eve and mallory love android: an analysis of android SSL (in)security. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 50–61. ACM (2012)
7. Fahl, S., Harbach, M., Oltrogge, M., Muders, T., Smith, M.: Hey, you, get off of my clipboard. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 144–161. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-39884-1\\_12](https://doi.org/10.1007/978-3-642-39884-1_12)
8. Felt, A.P., Ha, E., Egelman, S., Haney, A., Chin, E., Wagner, D.: Android permissions: user attention, comprehension, and behavior. In: Proceedings of the Eighth Symposium on Usable Privacy and Security, p. 3. ACM (2012)
9. Google: Android o behavior changes. <https://developer.android.com/preview/behavior-changes.html>. Accessed 24 July 2017
10. Google: Google privacy policy. <https://www.google.com/intl/en/policies/privacy/>. Accessed 24 July 2017
11. Haystack: The haystack project. <https://haystack.mobi/>. Accessed 24 July 2017
12. Hildebrandt, M., Gutwirth, S. (eds.): Profiling the European Citizen, Cross-Disciplinary Perspectives. Springer, Dordrecht (2008). <https://doi.org/10.1007/978-1-4020-6914-7>
13. Jia, Y.J., Chen, Q.A., Lin, Y., Kong, C., Mao, Z.M.: Open doors for bob and mallory: open port usage in android apps and security implications. In: 2017 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 190–203. IEEE (2017)
14. Kohno, T. (ed.): Proceedings of the 21th USENIX Security Symposium, Bellevue, WA, USA, 8–10 August 2012. USENIX Association (2012). [https://www.usenix.org/publications/proceedings/?f\[0\]=im\\_group\\_audience%3A334](https://www.usenix.org/publications/proceedings/?f[0]=im_group_audience%3A334)
15. Orthacker, C., Teufl, P., Kraxberger, S., Lackner, G., Gissing, M., Marsalek, A., Leibetseder, J., Prevenhueber, O.: Android security permissions – can we trust them? In: Prasad, R., Farkas, K., Schmidt, A.U., Lioy, A., Russello, G., Luccio, F.L. (eds.) MobiSec 2011. LNCS, vol. 94, pp. 40–51. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-30244-2\\_4](https://doi.org/10.1007/978-3-642-30244-2_4)
16. Peles, O., Hay, R.: One class to rule them all 0-day deserialization vulnerabilities in android. In: Proceedings of the 9th USENIX Conference on Offensive Technologies, p. 5. USENIX Association (2015)
17. Son, S., Kim, D., Shmatikov, V.: What mobile ads know about mobile users. In: NDSS (2016)
18. Stevens, R., Gibler, C., Crussell, J., Erickson, J., Chen, H.: Investigating user privacy in android ad libraries. In: Proceedings of the 2012 Workshop on Mobile Security Technologies (MoST) (2012)
19. The Guardian: Morality of mining for data in a world where nothing is sacred (2009). <https://www.theguardian.com/uk/2009/feb/25/database-state-ippaper>
20. Vallina-Rodriguez, N., Sundaresan, S., Razaghpanah, A., Nithyanand, R., Allman, M., Kreibich, C., Gill, P.: Tracking the trackers: towards understanding the mobile advertising and tracking ecosystem. CoRR abs/1609.07190 (2016). <http://arxiv.org/abs/1609.07190>
21. Vanrykel, E., Acar, G., Herrmann, M., Diaz, C.: Leaky birds: exploiting mobile application traffic for surveillance. In: Grossklags, J., Preneel, B. (eds.) FC 2016. LNCS, vol. 9603, pp. 367–384. Springer, Heidelberg (2017). [https://doi.org/10.1007/978-3-662-54970-4\\_22](https://doi.org/10.1007/978-3-662-54970-4_22)



# Towards Developing Network Forensic Mechanism for Botnet Activities in the IoT Based on Machine Learning Techniques

Nickolaos Koroniotis<sup>(✉)</sup>, Nour Moustafa, Elena Sitnikova,  
and Jill Slay

School of Engineering and Information Technology,  
University of New South Wales Canberra, Canberra, Australia  
n.koroniotis@student.adfa.edu.au,  
nour.moustafa@unsw.edu.au,  
{e.sitnikova, j.slay}@adfa.edu.au

**Abstract.** The IoT is a network of interconnected everyday objects called “things” that have been augmented with a small measure of computing capabilities. Lately, the IoT has been affected by a variety of different botnet activities. As botnets have been the cause of serious security risks and financial damage over the years, existing Network forensic techniques cannot identify and track current sophisticated methods of botnets. This is because commercial tools mainly depend on signature-based approaches that cannot discover new forms of botnet. In literature, several studies have conducted the use of Machine Learning (ML) techniques in order to train and validate a model for defining such attacks, but they still produce high false alarm rates with the challenge of investigating the tracks of botnets. This paper investigates the role of ML techniques for developing a Network forensic mechanism based on network flow identifiers that can track suspicious activities of botnets. The experimental results using the UNSW-NB15 dataset revealed that ML techniques with flow identifiers can effectively and efficiently detect botnets’ attacks and their tracks.

**Keywords:** Botnets · Attack investigation · Machine learning  
Internet of Thing (IoT)

## 1 Introduction

An increasingly popular new term, is the Internet of Things (IoT). The concept of IoT dates back to the early 1980s, where a vending machine selling Coca-Cola beverages located at the Carnegie Mellon University was connected to the Internet, so that its inventory could be accessed online to determine if drinks were available [32]. Today, the IoT is an umbrella term, covering a multitude of devices and technologies, that have both Internet capabilities, and serve some primary function, such as: home automation, including smart air conditioning system, smart fridge, smart oven and smart lamps, wearable devices (i.e., smart watch and fitness tracker), routers, healthcare, DVRs, smart cars, etc. In general, IoT can be viewed as a collection of devices with low processing

power and some form of network communication capabilities that allow a user to remotely access and use their services or view information collected by them [32, 33].

Recently, a number of malware have appeared that target IoT, as hackers and researchers alike have noticed that in the majority, IoT devices are mostly vulnerable to the simplest attacks, as displayed by Mirai, a botnet consisting of 100.000 infected “things” that in October 2016, attacked and took out a good portion of the Internet’s high-profile services such as Twitter and Netflix by doing a DDoS attack on Dyn (DNS provider) [35]. Since then, Mirai has slowly been divided into smaller botnets, and new botnets have risen, such as BrickerBot, which as its name implies “bricks” an IoT device (permanently disables it) and Hajime, a vigilante botnet which has been observed to infect devices targeted by Mirai, and “securing” the device, not allowing another malware to infect it. Clear methods have to be developed, in order to effectively mitigate and investigate at a forensic level IoT devices, as it is apparent that IoT is ever increasing in popularity, both by consumers (including companies) and hackers alike [34].

It is commonly known that the Internet is not a safe place, being full of security threats, with consequences ranging from mere inconvenience, to possible life-threatening scenarios. Amongst these threats, a family of cyber threats called Botnets, considered to be the one of the most destructive capabilities [20]. A botnet is defined as a set of internet-based appliances, which involves computer systems, servers and Internet of Thing (IoT) devices infected and managed by a common type of attacks, such as Denial of Service (DoS), DDoS, phishing attacks [15, 20]. Bots differ from other malware, in that they include a channel of communication with their creators, allowing them to issue commands to their network of bots (i.e., Zombies) and thus making botnets versatile when it comes to their functionality [1, 2].

Lately, there have been some prominent examples of botnets that harnessed the aggregated processing power of the IoT, and many infection routs have been revealed. One example is through the Internet, with portions of a botnet actively scanning the Internet for vulnerable devices, gaining access to them and then allowing a third device to infect the victim with the botnet malware [15]. Another is by using close proximity networks to gain direct communication with a device, infect it and allow the malware to propagate to the rest of the IoT devices in the immediate vicinity [16].

Different security controls have been used for defining botnet events, including Network forensic techniques and tools and intrusion detection and prevention systems. The existing techniques and tools basically use the principle of expert system which is generating specific rules in a blacklist and matching the upcoming network traffic against those rules. In this study, we investigate and analyze the role of ML techniques to construct a Network forensic technique based on network flow identifiers (i.e., source and destination IP address/ports and protocols). According to [21, 22], Network forensic is a branch of the digital forensics in order to monitor and inspect network traffic for defining the sources of security policy abuses and violations.

Machine learning techniques, learn and validate given network data for classifying legitimate and anomalous observations, have been utilized for building Network forensic techniques, but there are still two challenges should be addressed: producing high false alarm rates and defining the paths of attacks, in particular botnet events [4, 21–24]. Machine learning algorithms include clustering, classification, pattern recognition, correlation, statistical techniques [4, 23, 25]. In clustering techniques, network

traffic is separated into groups, based on the similarity of the data, without the need to pre-define these groups, while classification mechanisms learn and test patterns of network data associated with their class label [4].

The main contribution of this paper is the use of four classification techniques, so-called, Decision Tree C4.5 (DT), Association Rule Mining (ARM), Artificial Neural Network (ANN) and Naïve Bayes (NB) for defining and investigating the origins of botnets. The four techniques are used to recognize attack vectors, while the origins of attacks are linked with their flow identifiers as a Network forensic mechanism.

The rest of the paper is organized as follows. Section 2 discusses the background and related work then Sect. 3 explains the Network forensic architecture and its components. In Sect. 4, the experimental results are displayed and discussed and finally, the conclusion is given in Sect. 5.

## 2 Background and Previous Studies

This section discusses the background and related studies for the IoT, Botnets and Network forensics.

### 2.1 IoT

Even though IoT is slowly been assimilated in one form or another in everyday life, there is no doubt that there exist a number of issues with respect to security and privacy. One such example, is displayed by Ronen et al. [36]. The researchers investigate a possible attack vector for Philips Hue smart lamps, and a way that these IoT devices, which primarily use Zigbee as a communication protocol for the lamps to communicate with each other and their controllers. The attack vector they proposed and tested, takes advantage of exploits found in such devices and proposed that, under certain conditions (number of devices and relative distance), a malware can spread through a city, “jumping” from device to device, or even permanently disable (“brick”) them.

A more general point of view was adopted by Hossain et al. [33], who provided an analysis on open security problems in the IoT. They showed, through the literature, that IoT devices have been proven to be insecure, with researchers successfully compromising them with relative ease. Then, based on the IoT ecosystem, which is made up of: IoT devices, Coordinator, Sensor Bridge Device, IoT Service, Controller, they proceeded to highlight that conventional security and forensics mechanisms are inapplicable in IoT, as such devices are constrained in a number of ways (processing power, battery life, Network mobility), as-well-as diverse (IoT devices range from simple sensors, to complex Personal Computers) and research on those areas should become a priority. On the other hand, Pa et al. [37], made observations on the nature and type of IoT devices being actively scanned through the Internet and went on to propose an IoT specific Honeypot named IoTPOT. Their solution included an IoTBOX a collection of virtual IoT machines (Linux OS), which helps make the Honeypot appear as a

legitimate device. They observed a number of malware in action, most interestingly, they observed a repeating pattern, where a single host would perform the intrusion and information gathering process and the actual infection would be handled by a different host. Through the literature, it is evident that the field of IoT still being developed and as such various issues with this new and growing field are being discovered daily.

## 2.2 Botnets in IoT

In the literature, a variety of techniques that researchers utilize to understand Botnets and to study them has been observed. In their work, though not specifically related to IoT Botnets, Rahimian et al. [38] studied a Bot named Citadel. To specify, they employed several code analysis techniques to measure the similarities between Zeus and Citadel (Citadel being a “descendant” of the Zeus Bot) and in order to make the reverse engineering process faster they proposed a new approach named clone-based analysis, which attempts to identify parts of the malware’s code that originated from a different source, thus reducing the amount of code an analyst needs to review. Botnet scanning and identification was the main focus of Houmansadr and Borisov [39], who introduced BotMosaic, a tool following an architecture similar to Client-Server. Among other things BotMosaic, performs non-distorting network flow watermarking, for detecting IRC Botnets, which is the non-altering (towards network traffic content) process of “Marking” traffic so that it can be identified at a later time period. Although many techniques have been developed, in order to scan, fingerprint, identify and generally investigate a Botnet, as time moves forward, malware authors adapt and make their Bots more difficult to identify, this combined with the unique nature of the IoT, produces issues which need to be addressed.

Botnets are capable of launching a number of attacks, like Distributed Denial of Service attacks (DDoS), Keylogging, Phishing and Spamming, Identity theft and even other Bot proliferation [4]. Understandably, some research has been conducted in developing ways to detect and identify the presence of botnets in a network. One such way is the utilization of machine learning techniques on captured packets (Stored in files like pcap files) which are often grouped into network flows (Netflows), in an attempt to distinguish between legitimate user traffic and botnet traffic [5].

A botnet’s malware gets delivered to vulnerable targets through what is known as a propagation mechanism. Most commonly there exist two types of propagation, passive and active. Passive or self-propagation techniques rely on the bots themselves to actively scan the Internet for signs of vulnerable devices and then attempt to exploit the identified vulnerability, turning the vulnerable hosts into bots themselves [2, 3]. Passive propagation techniques, require users to access social media posts, storage media or websites that have been compromised and through user interaction, such as accessing URLs or other active parts of a website, download the malware (bot) to their machine, infecting it and making it part of the botnet [2, 3].

Various studies [17, 18, 21–25] have employed Machine learning techniques, to distinguish between normal and botnet network traffic and designing Network forensic techniques and tools. In their work, Roux et al. [17], created an intrusion detection system for IoT which takes into account wireless transmission data through probes. The

information is collected, which is relevant to signal strength and direction, determining that a signal originating from an unexpected direction, such as outside of the location that is being monitored, is deemed illegitimate. The classification of whether the observed signal indicates an attack or not is produced by a neural network. In another approach, Lin et al. [18], employed Artificial Fish Swarm Algorithm to produce the optimal feature set, which was then provided to a Support Vector Machine which detected botnet traffic. They reported a slight increase in accuracy when compared with Genetic Algorithms for feature selection, but produced great improvement time-wise. On the other hand, Greensmith [19], proposed the utilization of Artificial Immune Systems as a way of securing the IoT. They propose a combination of multiple AIS, in order to handle the heterogeneity of the IoT.

### 2.3 Network Forensics in IoT

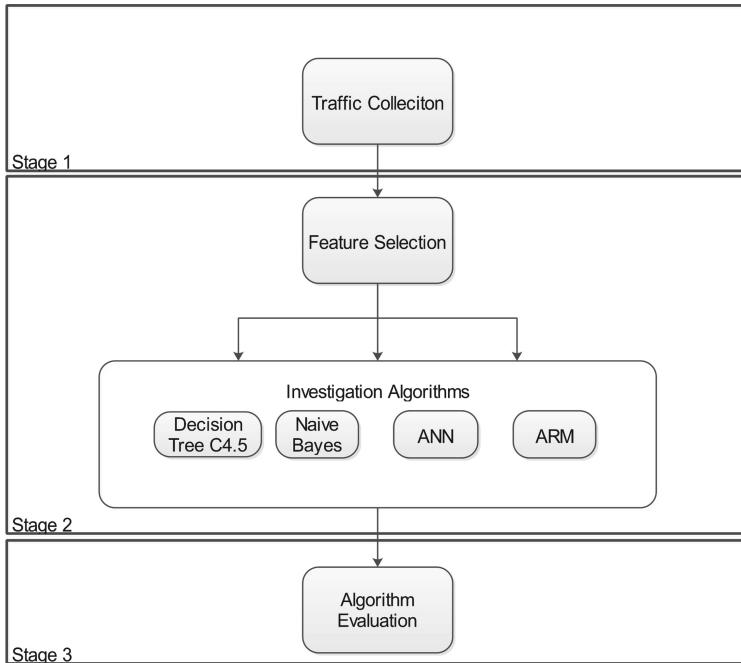
Network Forensics problem is governed by low availability of traces and evidence. For instance, Moustafa et al. [26] tackled the problem of the ever-increasing size of network log files, by using parallel execution through Hadoop's MapReduce. Bansal et al. [27] proposed their own generic framework for detecting Botnets, focusing on Network forensics techniques, such as packet capturing and inspection, which has the structure of a balanced framework, although it appears to be quite theoretical in nature.

Moustafa and Slay [28] employed an Artificial Neural Network for developing a distributed topology of DDoS inspectors residing in different networks, to detect DDoS attacks, based on timing and header inspection. Divakaran et al. [29] developed their own Framework for detecting such attacks (DDoS), by employing a regression model based on defined patterns. They did so, by grouping packets into network flows, and flows into sessions, based on timing of packet arrival, and then through regression, they identify anomalous patterns in traffic, also providing information on detection rates for different botnets.

On a different note all together, Wang et al. [30] proposed an attack detection and forensics technique to tackle the threat introduced by malware. In their approach, an attack detection model locates the presence of an incident after which, it cooperates with a honeypot to gather relevant information, which are then used by a forensics module to carry out an automated forensics procedure. Their work is interesting, as they effectively make the forensic process more active than what it usually is. As the Internet is incorporated in an ever-growing number of technologies that are gradually adopted by society, it should come as no surprise that Network forensics adopts a more integral role in investigating malicious activity.

## 3 Network Forensic Architecture and Components

The proposed Network forensics mechanism includes four components: (3.1) traffic collection, (3.2) network feature selection, (3.3) machine learning techniques, and (3.4) evaluation metrics, as depicted in Fig. 1 and explained below.



**Fig. 1.** Proposed Network forensic architecture

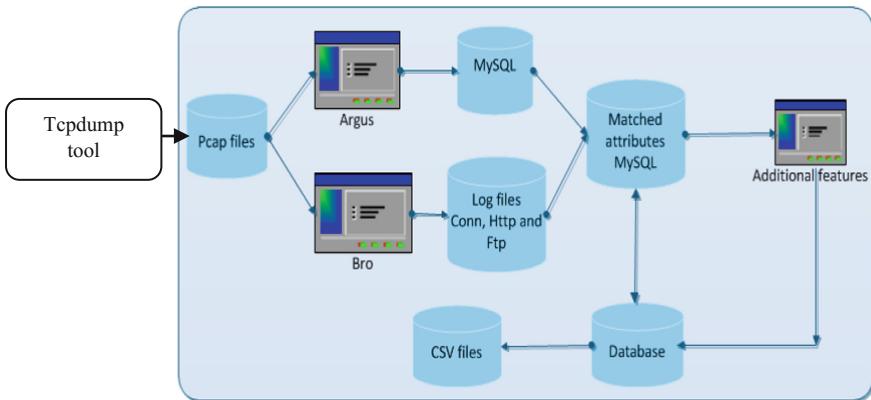
### 3.1 Traffic Collection

The immense volume of network traffic generated by today's networks, makes it necessary for a way to aggregate and summarize the captured packets, allowing for easier storage so that they can be used in construction of Network forensic mechanisms. Raw network packets are captured by using a `tcpdump` tool, which can access the network interface card to do so [14]. The next step is feature generation. By using tools like Bro and Argus, features are generated from the raw packets that were previously captured. The features in the UNSW-NB15 dataset were generated in this manner [14].

The network sniffing process needs to be conducted at key points of the network, particularly, at ingress routers, to make sure that the network flows that are gathered are relevant, which is determined by the source/destination IP address and protocol. This process helps, in investigating the origins of cyber-related incidents, and lowers the necessary processing time (Fig. 2).

### 3.2 Network Feature Selection Method

Feature selection is the method of adopting important/relevant attributes in a given dataset. Method of feature selection is classified into filter, wrapper and hybrid of the first two. A filter feature selection mechanism denotes selecting suitable features without the use of class label, while a wrapper one depends on ML techniques [7, 8]. An example of filtering methods for feature selection, is Information Gain (IG) and



**Fig. 2.** Collecting network features of UNSW-NB15 dataset [14]

Chi-square ( $\chi^2$ ). Wrapper methods study the performance of an algorithm that will be used in the end, as a criterion for selecting a suitable subset of the existing features. Intuitively, these methods split the features into subsets, and use these subsets to train the model that will be used after the pre-processing stage, using the error rate to score the subsets. Hybrid methods combine filter and wrapper methods to perform feature selection during the execution of machine learning algorithms.

We use the information gain feature mechanism as it is one of the simplest methods that can adopt relevant features in large-scale data, as in network datasets. More precisely, Information Gain (IG) selects features, by calculates the apparent reduction in entropy, when the feature to be selected is used to split the dataset.

### 3.3 Machine Learning Techniques

For the classification stage, we use the Weka tool [40] for applying four well-known machine learning algorithms. These algorithms are briefly described as follows.

- **Association Rule Mining (ARM)** [13] - is a classification algorithm, which is performed by generating rules of a form similar to  $\{V_1, V_2, \dots, V_n\} \Rightarrow \{C_1\}$ , where  $V_{1-n}$  are values of features and  $C_1$  is a class value.
- **Artificial Neural Network (ANN)** [10, 11] - is a classification model which was based on the idea of the human neurons [10, 11]. They usually are comprised of a number of neurons, which have multiple input flows and a single output, with some variants having multiple layers of units. The simplest form of an ANN is called a perceptron, taking the vector of attributes as input and classifying it.
- **Naïve Bayes (NB)** [12] - classifies a record  $R_1$  (collection of features) into a specific class  $C_2$ , if and only if the probability of that record to belong to that specific class, with respect to the record is greater than the probability of the record belonging to another class That is,  $P(C_2/R_1) > P(C_n/R_1)$ , with  $C_n$  being any class other than  $C_2$ .
- **Decision Tree C4.5 (DT)** [10] - is a classification algorithm which produces a tree-like structure to determine the class chosen for a record. The attributes are used

as the nodes of the tree and criteria are formulated leading from one node to the next, with the leaves being the Class value that is assigned to the record [10].

### 3.4 Evaluation Metrics

We utilize the confusion matrix [31] as a way of comparing the performance of the ML algorithms presented in the previous section. An example of a confusion matrix is given in Table 1. In simple terms, it is a table which depicts the possible outcomes of a classification, which in our case is either ‘1’, there was an attack detected or ‘0’ normal network traffic, against the actual values of the class feature already present in the evaluation (testing) dataset.

**Table 1.** Confusion matrix

	Actual negative	Actual positive
Predicted negative	TN	FP
Predicted positive	FN	TP

There are four conditions that can be shown in a confusion matrix, True Positive (TP), where the classifier has correctly identified the class feature and the value of that feature is positive (in our case there was an attack detected), True Negative (TN), similar to TP but the value of the class feature is negative (normal traffic), False Positive (FP), where the classifier identifies a record as an attack when, in actuality it is normal traffic and False Negative (FN), which incorrectly classifies an attack record as normal traffic.

By combining the TP, TN, FP, FN values we are able to create two metrics, namely **Accuracy** and **False Alarm Rate**, which we can use to evaluate the Classifiers. These two metrics are calculated as follows:

- **Accuracy** represents the probability that a record is correctly identified, either as attack, or as normal traffic. The calculation of Accuracy (Overall Success Rate) is  $OSR = (TN + TP)/(TP + FP + TN + FN)$
- **False Alarm Rate (FAR)** represents the probability that a record gets incorrectly classified. The calculation of the False Alarm Rate is  $FAR = FP/(FP + FN + TP + TN)$ .

## 4 Experimental Results and Discussions

### 4.1 Dataset Used for Evaluation and Feature Selection

In order to compare the four aforementioned algorithms, we used the UNSW-NB15 dataset designed at the Cyber Range Lab of the Australian Center of Cyber Security at UNSW Canberra [14]. We selected UNBS-NB 15, as it is one of the newest datasets in wide use today, thus providing accurate representations of both conventional (not malicious) network traffic, as-well-as a number of network attacks

performed by Botnets. The dataset was produced by making use of the IXIA PerfectStorm tool, which produced a mixture of legitimate user network traffic and attack traffic, with the latter being categorized into 9 groups, Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms.

A short description of these attacks is given here [41]. Fuzzers: where an attacker attempts to identify security weaknesses in a system, by providing large quantities of randomized data, expecting it to crash. Analysis, comprised of a variety of intrusion techniques targeting ports, email addresses and web scripts. Backdoor: a method of bypassing authentication mechanisms, allowing unauthorized remote access to a device. DoS: a disruption technique, which attempts to bring the target system in a state of non-responsiveness. Exploit: a combination of instructions, that take advantage of bugs in the code, leading the targeted system/network in a vulnerable state. Generic: an attack that attempts to cause a collision in a block-cipher using a hash function. Reconnaissance: an information gathering probe, usually launched before the actual attack. Shellcode: an attack during which carefully crafted commands are injected in a running application, through the net, allowing for further escalation by taking control of the remote machine. Worm: an attack based on a malware that replicates itself, thus spreading itself in a single or multiple host. The dataset is comprised of 49 features, including the class feature, and the portion of it that we will be making use, contains 257,673 (created by combining the training and testing datasets).

To test the classifiers, we performed Information Gain Ranking Filter (IG) for selecting the highest ten ranked features as listed in Table 2.

**Table 2.** UNSW-NB15 features selected with Information Gain

Ranking	Feature selected	Feature description
0.654	sbytes	Source to destination transaction bytes
0.491	dbytes	Destination to source transaction bytes
0.477	smean	Mean packet size transmitted by source
0.464	sload	Source bits per second
0.454	ct_state_ttl	
0.444	sttl	Source to destination time to live value
0.439	dttl	Destination to source time to live value
0.429	rate	
0.409	dur	Record total duration
0.406	dmean	Mean packet size transmitted by destination

## 4.2 Performance Evaluation of ML Algorithms

The confusion matrices of the four classification algorithms are listed in Tables 3, 4, 5 and 6 on the training and testing sets of the UNSW-NB15 dataset. The Weka tool was used for applying the four techniques using the default parameters with a 10-fold cross validation in order to effectively measure their performance.

Our experiments show that Decision Tree C4.5 Classifier was the best at distinguishing between Botnet and normal network traffic. This algorithm makes use of

**Table 3.** ARM confusion matrix

Normal	Attack	Prediction/Actual
31785	10894	Normal
12675	108654	Attack

**Table 4.** DT confusion matrix

Normal	Attack	Prediction/Actual
84607	8393	Normal
9058	155615	Attack

**Table 5.** NB confusion matrix

Normal	Attack	Prediction/Actual
84101	8899	Normal
61380	103293	Attack

**Table 6.** ANN confusion matrix

Normal	Attack	Prediction/Actual
2719	90281	Normal
2562	162111	Attack

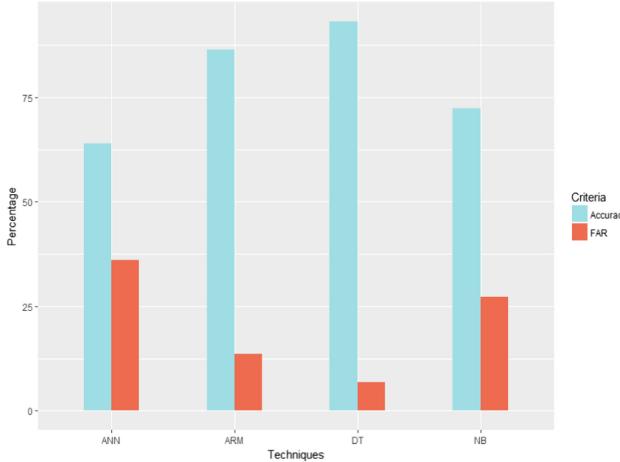
Information Gain, to pick the feature which best splits the data based on the classification feature, during construction of the tree and at every node. The test showed that DT had the highest accuracy out of all the algorithms that were tested at 93.23%, and the lowest FAR at 6.77%.

ARM was the second-best classifier, having an accuracy of close to 86% and FAR just over twice that of the DT. The Naïve Bayes classifier, which relies on probability to classify records in classes was third, with 20% less accuracy and close to 21% more false alarms than the DT. Finally, the Artificial Neural Network was the least accurate out of the four algorithms that we tested, with accuracy and false alarm rate for this classifier showing a 30% differentiation from the C4,5 algorithm (Table 7).

**Table 7.** Performance evaluation of four techniques

Classifier	Accuracy	FAR
ARM	86.45%	13.55%
DT	93.23%	6.77%
NB	72.73%	27.27%
ANN	63.97%	36.03%

A similar comparison of Machine Learning performance was conducted on the KDD99 dataset, arguably one of the most popular datasets still in use for security related work, like evaluating Network Intrusion Detection Systems [41] (Fig. 3).



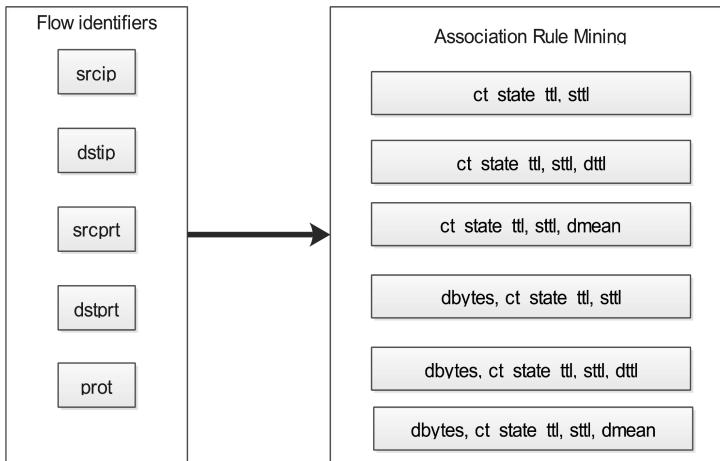
**Fig. 3.** Accuracy vs. FAR of four classifiers

By combining the identifiers of a network flow, with the corresponding condition Label, which depicts the result of the classification techniques mentioned previously that classify a record under “attack” or “normal”, the tracking of attack instances becomes possible. An example of the final form of the dataset is given in Table 8, which provides a number of flows and their classification label, taken from the UNSW-NB15 dataset. The network forensic technique that this paper illustrates, can assist network administrators, security experts or even law enforcement, to identify, track, report and even mitigate security incidents that threaten the security of their network.

**Table 8.** ANN confusion matrix

Classifier	Accuracy	FAR
ARM	92.75%	—
DT	92.3%	11.71%
NB	95%	5%
ANN	97.04%	1.48%

To produce the results depicted in Table 8, first the combination of flow information and Association Rule Mining needs to be performed, as shown in Fig. 4. The produced rules indicate the type of attack and the means by which it was identified (features used in the rule), and later by combining that information with a row



**Fig. 4.** Combining flow data with rules.

containing source/destination IP address/ Port numbers and protocol used, it becomes possible to attribute specific botnet activities to specific hosts. Such association between flow information and rule, can be of vital importance in monitoring botnet activities, through identifying infected hosts and showing their actions over time (Table 9).

**Table 9.** Flow identifiers associated with actual label for investigating attacks

srcip	sport	dstip	dsport	proto	Label
149.171.126.14	179	175.45.176.3	33159	tcp	0
149.171.126.18	1043	175.45.176.3	53	udp	0
175.45.176.3	46577	149.171.126.18	25	tcp	1
149.171.126.15	1043	175.45.176.3	53	udp	0
175.45.176.2	16415	149.171.126.16	445	tcp	1

## 5 Conclusions

This paper discusses the role of machine learning techniques for identifying and investigating botnets. There are four ML techniques of DT, ANN, NB and ANN machine are evaluated on the USNW-NB15 dataset. The accuracy and false alarm rate of the techniques are assessed, and the results revealed the superiority of the DT compared with the others. The best machine learning techniques and flow identifiers of source/destination IP addresses and protocols can effectively and efficiently detect botnets and their origins as Network forensic mechanism.

**Acknowledgements.** Nickolaos Koroniots would like to thank the Commonwealth's support, which is provided to the aforementioned researcher in the form of an Australian Government Research Training Program Scholarship.

## References

1. Silva, S.S.C., Silva, R.M.P., Pinto, R.C.G., Salles, R.M.: Botnets: a survey. *Comput. Netw.* **57**(2), 378–403 (2013)
2. Khattak, S., Ramay, N.R., Khan, K.R., Syed, A.A., Khayam, S.A.: A taxonomy of Botnet behavior, detection, and defense. *IEEE Commun. Surv. Tutor.* **16**(2), 898–924 (2014)
3. Negash, N., Che, X.: An overview of modern Botnets. *Inf. Secur. J.: Glob. Perspect.* **24**(4–6), 127–132 (2015)
4. Amini, P., Araghizadeh, M.A., Azmi, R.: A survey on Botnet: classification, detection and defense. In: 2015 International Electronics Symposium (IES), pp. 233–238. IEEE (2015)
5. Goodman, N.: A survey of advances in Botnet technologies. arXiv preprint [arXiv:1702.01132](https://arxiv.org/abs/1702.01132) (2017)
6. Sheen, S., Rajesh, R.: Network intrusion detection using feature selection and Decision tree classifier. In: TENCON 2008-2008 IEEE Region 10 Conference. IEEE (2008)
7. Chandrashekhar, G., Sahin, F.: A survey on feature selection methods. *Comput. Electr. Eng.* **40**(1), 16–28 (2014)
8. Jović, A., Brkić, K., Bogunović, N.: A review of feature selection methods with applications. In: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE (2015)
9. Bhavsar, Y.B., Waghmare, K.C.: Intrusion detection system using data mining technique: support vector machine. *Int. J. Emerg. Technol. Adv. Eng.* **3**(3), 581–586 (2013)
10. Area, S., Mesra, R.: Analysis of bayes, neural network and tree classifier of classification technique in data mining using WEKA (2012)
11. Sebastian, S., Puthiyidam, J.J.: Evaluating students performance by artificial neural network using weka. *Int. J. Comput. Appl.* **119**(23) (2015)
12. Xiao, L., Chen, Y., Chang, C.K.: Bayesian model averaging of Bayesian network classifiers for intrusion detection. In: 2014 IEEE 38th International Computer Software and Applications Conference Workshops (COMPSACW), pp. 128–133. IEEE (2014)
13. Moustafa, N., Slay, J.: The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems. In: 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS). IEEE (2015)
14. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: Military Communications and Information Systems Conference (MilCIS), 2015. IEEE (2015)
15. Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., Rossow, C.: IoTPOT: analysing the rise of IoT compromises. *EMU* **9**, 1 (2015)
16. Ronen, E., Shamir, A., Weingarten, A.O., O'Flynn, C.: IoT goes nuclear: creating a ZigBee chain reaction. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 195–212 (2017)
17. Roux, J., Alata, E., Auriol, G., Nicomette, V., Kaâniche, M.: Toward an intrusion detection approach for IoT based on radio communications profiling. In: 13th European Dependable Computing Conference (2017)
18. Lin, K.C., Chen, S.Y., Hung, J.C.: Botnet detection using support vector machines with artificial fish swarm algorithm. *J. Appl. Math.* **2014**, 9 (2014)

19. Greensmith, J.: Securing the Internet of Things with responsive artificial immune systems. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 113–120. ACM (2015)
20. Pijker, J., Vranken, H.: The role of internet service providers in botnet mitigation. In: Intelligence and Security Informatics Conference (EISIC), 2016 European. IEEE (2016)
21. Wang, X.-J., Wang, X.: Topology-assisted deterministic packet marking for IP traceback. *J. China Univ. Posts Telecommun.* **17**(2), 116–121 (2010)
22. Khan, S., Gani, A., Wahab, A.W.A., Shiraz, M., Ahmad, I.: Network forensics: review, taxonomy, and open challenges. *J. Netw. Comput. Appl.* **66**, 214–235 (2016)
23. Moustafa, N., Slay, J., Creech, G.: Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. *IEEE Trans. Big Data*
24. Prakash, P.B., Krishna, E.S.P.: Achieving high accuracy in an attack-path reconstruction in marking on demand scheme. *i-Manager's J. Inf. Technol.* **5**(3), 24 (2016)
25. Sangkatsanee, P., Wattanapongsakorn, N., Charnsripinyo, C.: Practical real-time intrusion detection using machine learning approaches. *Comput. Commun.* **34**(18), 2227–2235 (2011)
26. Moustafa, N., Creech, G., Slay, J.: Big data analytics for intrusion detection system: statistical decision-making using finite dirichlet mixture models. In: Palomares Carrascosa, I., Kalutarage, H.K., Huang, Y. (eds.) *Data Analytics and Decision Support for Cybersecurity*. DA, pp. 127–156. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-59439-2\\_5](https://doi.org/10.1007/978-3-319-59439-2_5)
27. Bansal, S., Qaiser, M., Khatri, S., Bijalwan, A.: Botnet Forensics Framework: Is Your System a Bot. In: 2015 Second International Conference on Advances in Computing and Communication Engineering, Dehradun, 2015, pp. 535–540 (2015)
28. Moustafa, N., Slay, J.: A hybrid feature selection for network intrusion detection systems: central points. arXiv preprint [arXiv:1707.05505](https://arxiv.org/abs/1707.05505) (2017)
29. Divakaran, D.M., Fok, K.W., Nevat, I., Thing, V.L.L.: Evidence gathering for network security and forensics. *Digit. Investig.* **20**(S), S56–S65 (2017)
30. Wang, K., Du, M., Sun, Y., Vinel, A., Zhang, Y.: Attack detection and distributed forensics in machine-to-machine networks. *IEEE Netw.* **30**(6), 49–55 (2016)
31. Moustaf, N., Slay, J.: Creating novel features to anomaly network detection using darpa-2009 data set. In: Proceedings of the 14th European Conference on Cyber Warfare and Security. Academic Conferences Limited (2015)
32. Rose, K., Eldridge, S., Chapin, L.: The Internet of Things: an overview (2015)
33. Hossain, M.M., Fotouhi, M., Hasan, R.: Towards an analysis of security issues, challenges, and open problems in the internet of things. In: 2015 IEEE World Congress on Services, New York City, NY, pp. 21–28 (2015)
34. Shattuck, J., Boddy, S.: Threat Analysis Report DDoS's Latest Minions: IoT Devices. F5 LABS, vol. 1 (2016)
35. Schneier, B.: Botnets of things. *MIT Technol. Rev.* **120**(2), 88–91 (2017). Business Source Premier, EBSCOhost. Accessed 24 Aug 2017
36. Ronen, E., O'Flynn, C., Shamir, A., Weingarten, A.-O.: IoT goes nuclear: creating a ZigBee chain reaction. In: Cryptology ePrint Archive, Report 2016/1047 (2016)
37. Pa, Y.M.P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., Rossow, C.: IoTPOT: analysing the rise of IoT compromises. In: Francillon, A., Ptacek, T. (eds.). *Proceedings of the 9th USENIX Conference on Offensive Technologies (WOOT 2015)*. USENIX Association, Berkeley, CA, USA, p. 9 (2015)
38. Rahimian, A., Ziarati, R., Preda, S., Debbabi, M.: On the reverse engineering of the citadel botnet. In: Danger, J.-L., Debbabi, M., Marion, J.-Y., Garcia-Alfaro, J., Zincir Heywood, N. (eds.) *FPS -2013. LNCS*, vol. 8352, pp. 408–425. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-05302-8\\_25](https://doi.org/10.1007/978-3-319-05302-8_25)

39. Houmansadr, A., Borisov, N.: BotMosaic: collaborative network watermark for the detection of IRC-based botnets. *J. Syst. Softw.* **86**(3), 707–715 (2013). ISSN 0164-1212
40. Weka tool. <http://www.cs.waikato.ac.nz/ml/weka/>. Accessed Aug 2017
41. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Inf. Secur. J.: Glob. Perspect.* **25**(1–3), 18–31 (2016)



# Performance Comparison of Distributed Pattern Matching Algorithms on Hadoop MapReduce Framework

C. P. Sona<sup>(✉)</sup> and Jaison Paul Mulerikkal

Súnya Labs, Rajagiri School of Engineering and Technology, Kochi, India  
sonacp@rajagiritech.edu.in

**Abstract.** Creating meaning out of the growing Big Data is an insurmountable challenge data scientists face and pattern matching algorithms are great means to create such meaning from heaps of data. However, the available pattern matching algorithms are mostly tested with linear programming models whose adaptability and efficiency are not tested in distributed programming models such as Hadoop MapReduce, which supports Big Data. This paper explains an experience of parallelizing three of such pattern matching algorithms, namely - Knuth Morris Pratt Algorithm (KMP), Boyer Moore Algorithm (BM) and a lesser known Franek Jennings Smyth (FJS) Algorithm and porting them to Hadoop MapReduce framework. All the three algorithms are converted to MapReduce programs using key value pairs and experimented on single node as well as cluster Hadoop environment. The result analysis with the Project Gutenberg data-set has shown all the three parallel algorithms scale well on Hadoop environment as the data size increases. The experimental results prove that KMP algorithm gives higher performance for shorter patterns over BM, and BM algorithm gives higher performance than KMP for longer patterns. However, FJS algorithm, which is a hybrid of KMP and Boyer horspool algorithm which is advanced version of BM, outperforms both KMP and BM for shorter and longer patterns, and emerges as the most suitable algorithm for pattern matching in a Hadoop environment.

**Keywords:** Pattern matching · Hadoop · MapReduce · Big Data  
Knuth Morris Pratt Algorithm · Boyer Moore Algorithm  
Franek Jennings Smyth Algorithm

## 1 Introduction

Big Data is defined as a large collection of data-sets that grow exponentially with time. It is generally understood that the four characteristics of Big Data are volume, velocity, variety and veracity [13]. This varied large volume of data can be from applications like the Large Hadron Collider of CERN or from the human genome project. It could also be the data generated by jet engines, which

can generate up to 10 terabytes of data in 30 min of flight time [5]. Another example is the New York Stock Exchange which generates about 1 terabyte of new trade data per day [14].

The Big Data can be structured, semi-structured or unstructured. The data which is structured and semi-structured can be addressed using the traditional data management tools but unstructured data still remains unsolved using traditional methods. The efficient processing tool that can deal with Big Data is Hadoop MapReduce framework developed by Doug Cutting [17]. It is systematically designed to process Big Data in a scalable way through distributed processing. The Hadoop Distributed File System (HDFS) creates the distributive environment that is required for the parallel processing of data. The Mapper and Reducer functions help to split and parallelize the work for faster processing of data.

Pattern matching algorithms are good candidates to decipher insights from Big Data. They are also known as string matching algorithms. These are essential classes of string algorithms which help discover one or all existences of the string within an enormous group of text [3]. It is a solution for many real world problems. Many applications such as twitter analysis, information retrieval, sentimental analysis and DNA analysis use pattern matching algorithms at different stages of processing. For example, protein link prediction using the DNA genes requires a stage where a good pattern matching algorithm is required to match the DNA pattern and take a count of matched DNA for further processing. Normal prediction requires high execution time due to processing of more than fifty thousands of DNA pattern. This execution time can be improved using effective pattern matching algorithm which works well on distributed environment [12]. The efficiency varies with applications as well as different parameters like pattern length, data-set etc.

Certain pattern matching algorithms such as Knuth Morris Pratt Algorithm (KMP), Brute Force Algorithm and Boyer Moore Algorithm (BM) have proven to be some of the optimal solutions for such applications [7, 18]. There were several attempts to parallelize KMP, BM and Brute Force algorithms for small sets of data. However those efforts found it difficult to use these parallel algorithms for large sets of data by distributing Big Data among many nodes. Hadoop becomes a natural candidate to overcome this difficulty.

This study is focused on creating parallelized versions of these three algorithms - viz., KMP, BM and FJS - to work with Hadoop MapReduce framework, so as to scale well with Big Data to produce increased performance. The experiments have been carried out on single node as well as Hadoop MapReduce setups with different sizes of data-sets and lengths of patterns using Project Gutenberg textual data-set. FJS algorithm proves to be the most efficient algorithm on Hadoop MapReduce framework for shorter as well as longer patterns. Other inferences are explained in detail in the result analysis section.

In this paper, Sect. 2 will discuss background studies and related works. Section 3 gives a brief description about pattern matching and algorithms used.

Section 4 describes the experimental setup and followed by result analysis at Sect. 4.6. Section 5 gives the conclusion and future scope.

## 2 Related Works

In the past there were many attempts [9, 10, 16] to parallelize pattern matching algorithms in distributed environments. Many attempts succeeded in dealing with small sets of data but either never tried with large sets of data or were confronted with road blocks when chose to deal with it.

Diwate and Alaspurkar [11] conducted linear experiments on different pattern matching algorithms which gave the conclusion that the KMP algorithm is more effective than the BM and Brute Force algorithm. They found out that time performance of exact string pattern matching can be greatly improved if KMP algorithm is used.

Alzoabi et al. [16] proposed a parallel KMP algorithm using MPI programming which give better improvement in execution time. However they could not find an efficient way to split the data when it became large. Cao and Wu [10] have also parallelized the KMP algorithm using MPI programming but it started to show communication errors when the number of processes exceeded 50 or so.

Kofahi and Abusalama [9] have proposed a framework for distributed pattern matching based on multi-threading using java programming. The framework addresses the problems in splitting the textual data sets, but it is not efficient when large number of smaller text files are processed.

Sardjono and Al Kindhi [15] proposed that performance measurement of pattern matching on large sets of Hepatitis C Virus Sequence DNA data showed that Boyer Moore is efficient when comes to minimum shift technique whereas Brute Force algorithm has higher accuracy for pattern matching or finding a match. The study also proved that either KMP or BM algorithm can be chosen as appropriate algorithm according to the pattern length. The disadvantage was that they did not conduct the study in a distributed environment.

Ramya and Sivasankar [2] explains the efforts to port KMP algorithm to Hadoop MapReduce framework and proved that it is possible. However, they have done experiments for only single occurrence of pattern.

Franek et al. have introduced a new linear pattern matching algorithm, which is a hybrid version of KMP and BM. It uses the good features of both KMP and BM for execution and it is explained in [19]. This paper proves that their algorithm, which is generally know as FJS algorithm, has better execution time than most other pattern matching algorithms available, including KMP, BM and Brute Force.

It is in this context, that experiments focus to find an optimal algorithm that can work well with Hadoop MapReduce framework which will be helpful in dealing with Big Data applications. Drawing inspiration from the above literature survey, KMP, BM and FJS algorithms were chosen to test on Hadoop MapReduce framework and to compare the efficiency of pattern matching algorithms on a distributed environment.

**Algorithm 1.** Knuth Morris Pratt algorithm

---

```

class Mapper
method Initialize
H = new AssociativeArray
method Map(docid id, doc d)
for all term t in doc d
If t statifies KMP.Prefix(i) do
KMP.SearchPattern(t,p,Prefix(i))
H[t] = H[t] + 1
Emit(term t, count 1)
for all term t in H do
Emit(term t, count H[t])
class KMP
method Initialize
p ← Pattern
method ComputePrefix(p,i,j)
return Prefix(i)
method SearchPattern(t,p,Prefix(i))
return pattern(t,id)
class Reducer
method Reduce(term t, counts [c1, c2,...])
sum = 0
for all count c in [c1, c2,...] do
sum = sum + c
Emit(term t, count sum)

```

---

### 3 KMP, BM and FJS Algorithms and Their MapReduce Versions

Pattern or string matching can include single pattern algorithms, algorithms using a finite set of patterns and algorithms using infinite number of patterns. Single pattern algorithms used here includes KMP algorithm, BM algorithm and some improved algorithms which includes FJS algorithm. BM algorithm is considered as the bench mark algorithm for pattern matching [6]. KMP is considered as first linear time string-matching algorithm So It was important to check its efficiency in distributed environment. FJS algorithm which is the hybrid combination of KMP and BM has also chosen for experiments since it was proved to be efficient in linear implementations as reported in the previous section.

#### 3.1 Knuth Morris Pratt Algorithm

The KMP algorithm was developed by Knuth and Pratt, and independently by Morris. The KMP algorithm uses prefix table for string matching to avoid backtracking on string for redundant checks. It has been reported that it works well on shorter patterns [7]. The KMP algorithm is explained in [8] and its MapReduce version is given at Algorithm 1. The KMP matcher performs the shifts

**Algorithm 2.** Boyer Moore algorithm

---

```

class Mapper
method Initialize
  H = new AssociativeArray
  method Map(docid id, doc d)
    for all term t in doc d
      If t statifies BM.Search() do
        H[t] = H[t] + 1
      else BM.Badrule()
        Emit(term t, count 1)
    for all term t in H do
      Emit(term t, count H[t])
  class BM
    method Initialize
      p ← Pattern
      method Search(p, i, j)
        while ( P.charAt(j) == T.charAt(i0+j) )
          j—
        return P with corresponding term t
      method BM.Badrule()
        return i0 = i0 + j - lastOcc[T.charAt(i0+j)]
    class Reducer
      method Reduce(term t, counts [c1, c2,...])
        for all count c in [c1, c2,...] do
          sum = sum + c
        Emit(term t, count sum)

```

---

while performing string matching. While porting this algorithm to MapReduce programming paradigm, the document ID and document contents are selected as key, value pairs. The result of which emits the term of document contents containing the required pattern with number of occurrences.

### 3.2 Boyer Moore Algorithm

In BM algorithm [4], the string check is done from right end of the string. It uses bad character shift table and good suffix table. This is generally used for DNA analysis. As reported in [4] BM algorithm works well for long pattern lengths.

The MapReduce adaptation of this algorithm is given at 2. Here in the mapper phase the document ID and contents of document is selected as key value pairs. The mapper phase emits the terms in documents which satisfies the BM.Search() where it tries to match from the end of the string and if match position is 0 then jump ahead characters. The search continues based on for each character perform right-to-left scan. The bad character rule for each character for rightmost occurrence of character in pattern  $p$  is assigned to be zero if character does not occur in  $p$ . The terms satisfying the predicate are recorded

**Algorithm 3.** FJS algorithm

---

```

1: class Mapper
2: method Initialize
3: H = new AssociativeArray
4: method Map(docid id, doc d)
5: for all term t in doc d
6: If t statisfies FJS.Search() do
7: class FJS
8: method Initialize
9: p ← p(1..x)
10: t ← t(1..y)
11: method Search(p, t, doc d)
12: if x < i then return
13:     i' ← x
14: end if
15: if i' < n then
16:     x' ← x-1
17: end if
18: sundayshift.
19: x[i'] ← p[m].
20: i ← x-1.
21: KMP-Match(x, t)
22: return Pattern
23: class Reducer
24: method Reduce(term t, counts [c1, c2,...])
25: sum = 0
26: for all count c in [c1, c2,...] do
27: sum = sum + c
28: Emit(term t, count sum)

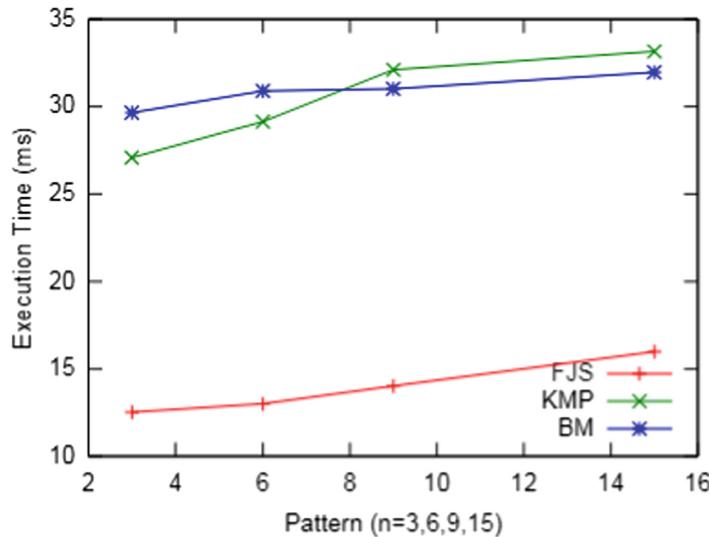
```

---

in a associative array. At reducer phase the term containing the pattern and its number of occurrences are emitted as list of key-value pairs.

### 3.3 Franek Jennings Smyth Algorithm

The FJS algorithm is a hybrid algorithm of KMP and BM. It uses KMP algorithm if it finds a partial match, else it uses the simplified version of BM method with help of sunday-shift. The algorithm only use bad-character shift for computing. Pre-processing phase prepare the bad character shift value and that table used during the searching phase of algorithm. The algorithm is explained at [19] and its MapReduce version is presented at Algorithm 3. The mapper function reads the contents from the documents emits the key-value pair containing the terms and count which satisfies the FJS predicate. The reducer function outputs the terms its total number of occurrences.



**Fig. 1.** Execution time of pattern matching of different lengths of patterns on 1 GB

## 4 Performance Analysis

### 4.1 Experimental Setup

The experiments are conducted on three different configurations as explained at Sects. 4.2, 4.3 and 4.4. The configurations include experiment on single node as well as cluster Hadoop implementations. The initial studies were conducted on a single node commodity machine installed with Hadoop for single node. It was then extended to a single node server machine with Hadoop for single node installation. Final experiments were conducted on a commodity cluster installed with Hadoop cluster version. Hadoop version 2.2.0 and Eclipse IDE are used. In all the cases (Sects. 4.2, 4.3 and 4.4), the data-set used was the open data available at Project Gutenberg [1]. An average of execution time is taken from 3 consecutive runs of the program for all cases.

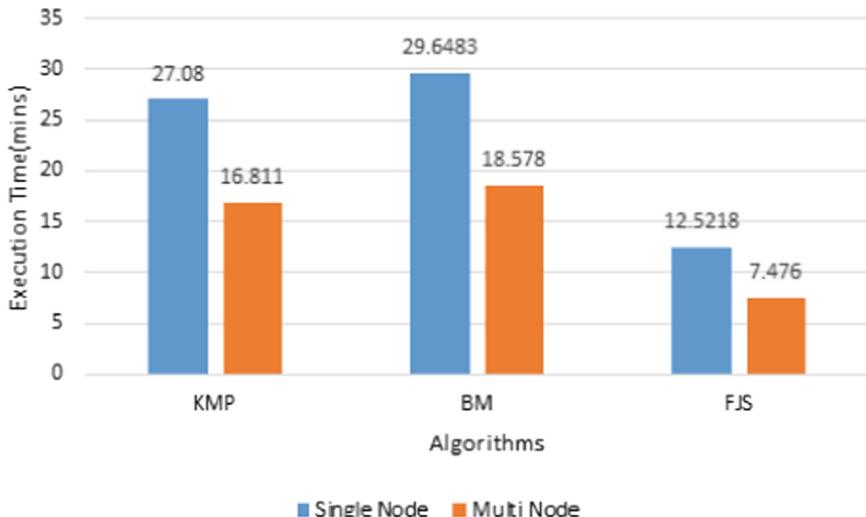
### 4.2 Single Node on Commodity Machine

The Single node Hadoop configuration was first tested on Intel Core i3-2120 machine with 8 GB RAM. The machines ran Ubuntu 16.04 OS and used Eclipse IDE as programming framework. Pattern matching experiment on dataset size of 1 GB were carried out. Each algorithm was evaluated with an increase in pattern length from 3 term to 15 term which was performed on 1 GB dataset.

### 4.3 Single Node on Server Machine

The single node Hadoop implementation was done on an Intel Xeon E5-2650 V3 server machine with a RAM of 32 GB. The OS used was Debian 8 (Jessie).

Pattern matching experiments of varying sizes of data were carried out on this configuration.



**Fig. 2.** Execution time analysis of algorithms on single node setup configuration of Sect. 4.2 and multi-node Sect. 4.4

#### 4.4 Multi-node on Commodity Machine

This configuration consists of three Intel Core i3-2120 machines (similar to the machine at Sect. 4.3) each with 8 GB RAM. The multi-node Hadoop system was introduced by configuring one of the systems as master node. The same system also runs a slave instance. The other two machines run one each slave instances. So, the multi-node setup consists of one master and three slave instances. All nodes run Ubuntu 16.04 OS and Eclipse IDE is used for programming development. Pattern matching experiments on data-set size of 1 GB were carried out on this multi-node configuration.

#### 4.5 Details of Experiments

The data-set from Project Gutenberg which contains large collection of small text data was pre-processed and was loaded to HDFS. In the first phase of experiment, the pattern matching was performed on Hadoop single node. Each algorithm was tested on data-set of size 1 GB to 3 GB.

All occurrence of patterns with matching index line was stored in an output file. The execution time of three pattern matching algorithms (i.e. KMP, BM and FJS) were noted and average is reported in Fig. 3.



**Fig. 3.** Execution time analysis of algorithms on single node configuration Sect. 4.3

In the second stage of experimental study, each algorithm is processed on a data-set of 1 GB with different pattern length ranging from 3 term pattern to 15 term. The results are shown in Fig. 1.

In the final sets of experiments, the Hadoop multi-node cluster with 1 master node and 3 slave nodes are used. Pattern matching is performed on 1 GB of data. The results of these experiments are shown in Fig. 2.

#### 4.6 Result Analysis and Performance Comparison

Figure 3 shows the performance of algorithms in terms of mapper time (T1), reducer time (T2) and execution time (T3). Our experiments prove that all the three algorithms under consideration, scale well with regards to increase in data on Hadoop MapReduce framework with almost linear progression for T1, T2 and T3 as shown in Fig. 3 on a single node server configuration as explained in Sect. 4.3. This suggests that these algorithms can be successfully parallelized using Hadoop MapReduce framework to analyze increasing data, or in other words, Big Data.

The results of pattern matching experiments performed on a uniform 1 GB data with different pattern sizes ranging from 3 terms to 15 terms using single node configuration as experimented in Sect. 4.2 are reported in Fig. 1. Figure 1 has corroborated previously reported trends of linear KMP and BM algorithms on a Hadoop MapReduce framework as explained below.

It was reported on [7, 18] that BM algorithm shows better performance than KMP algorithm for longer pattern lengths using linear programming models. Figure 1 proves that this is also true with case of Hadoop MapReduce framework

versions of these algorithms. It was also reported in [7] that KMP shows better performance than BM on shorter pattern length using linear algorithms. Our results also prove that this advantage of KMP using shorter pattern length is replicated in a Hadoop MapReduce framework as shown in Fig. 1. However, the most striking insight is that as expected, the FJS algorithm showed much faster pattern matching execution time for both shorter and longer patterns of length on textual data for all different sizes of data comparing KMP and BM using Hadoop MapReduce framework. This proves that FJS algorithm is the optimal solution for pattern matching application for Big Data on Hadoop MapReduce framework.

Figure 2 shows the scaling of all the three algorithms moving from single node Hadoop setup to a multi-node Hadoop setup. This is done using similar machines in single node as well as multi-node configurations as explained in Sects. 4.2 and 4.4 with a standard 3 term pattern. Figure 2 shows around 40% performance improvement for all algorithms from single node (Sect. 4.2) to multi-node (refmulti-com) installation of Hadoop. This result safely proves that scaling of performance in a Hadoop environment is possible as the number of compute nodes increases for the above algorithms using MapReduce programming framework. FJS emerges as the most suitable candidate in this scenario as well.

## 5 Conclusion and Future Scope

The experiments have clearly proved that Hadoop MapReduce versions of KMP and BM algorithms work efficiently on shorter and longer patterns respectively in a distributed environment. The study shows that FJS is the optimum pattern matching algorithm in a Hadoop distributed environment compared to KMP and BM. It indicates the potential of FJS algorithm to be a solution for many real time Big Data applications like text analytics, information retrieval and DNA pattern matching.

The future scope of this work is an enhancement to FJS algorithm on Hadoop MapReduce framework. For enhancing the FJS method, a Hash Join function can be introduced. The main benefit of using the hash function will be to reduce the number of character comparisons performed by FJS algorithm in each attempt. Thus, it will reduce the required comparison time. The enhanced FJS algorithm can be explored with its feasibility in different system configurations using Hadoop MapReduce framework. The potential of Enhanced FJS algorithm can be further explored using a real-time Big Data application such as twitter data analysis.

**Acknowledgement.** This work was completed successfully using the infrastructure support provided by Šunya Labs, Rajagiri School of Engineering and Technology, India.

## References

1. Project gutenberg. <https://www.gutenberg.org/>
2. Ramya, A., Sivasankar, E.: Distributed pattern matching and document analysis on big data using Hadoop MapReduce model. In: International Conference on Parallel and Distributed Grid Computing (2014)
3. Al-Mazroi, A.A., Rashid, N.A.A.: A fast hybrid algorithm for the exact string matching problem. *Am. J. Eng. Appl. Sci.* **4**(1), 102–107 (2011)
4. Boyer, R.S.: A fast string searching algorithm. *Commun. Assoc. Comput. Mach.* **20**, 762–772 (1977)
5. Finnegan, M.: Boeing 787s to create half a terabyte of data per flight, says Virgin Atlantic. <http://www.computerworlduk.com/data/>. Accessed 12 Sep 2017
6. Hume, A., Sunday, D.: Fast string searching. *Softw.: Pract. Exp.* **21**(11), 1221–1248 (1991)
7. Al-Khamaiseh, K., ALShagarin, S.: A survey of string matching algorithms. *Int. J. Eng. Res. Appl.* **4**, 144–156 (2014). ISSN 2248–9622
8. Knuth, D.E., Morris, J.H., Pratt, V.R.: Fast pattern matching in strings. *SIAM J. Comput.* **6**, 323–350 (1977)
9. Kofahi, N., Abusalama, A.: A framework for distributed pattern matching based on multithreading. *Int. Arab J. Inf. Technol.* **9**(1), 30–38 (2012)
10. Cao, P., Wu, S.: Parallel research on KMP algorithm. In: International Conference on Consumer Electronics, Communications and Networks (CECNet) (2011)
11. Diwate, M.R.B., Alaspurkar, S.J.: Study of different algorithms for pattern matching. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **3**, 1–8 (2013). ISSN 2277 128X
12. Rajesh, S., Prathima, S., Reddy, L.S.S.: Unusual pattern detection in DNA database using KMP algorithm. *Int. J. Comput. Appl.* **1**(22), 1–5 (2010)
13. Singh, S., Singh, N.: Big data analytics. In: 2012 International Conference on Communication, Information and Computing Technology (ICCICT), 13230053, IEEE, October 2012
14. Singh, A.: New York stock exchange oracle exadata - our journey. <http://www.oracle.com/technetwork/database/availability/index.html>. Accessed 12 Sep 2017
15. Sardjono, T.A., Al Kindhi, B.: Pattern matching performance comparisons as big data analysis recommendations for hepatitis C virus (HCV) sequence DNA. In: International Conference on Artificial Intelligence, Modelling and Simulation (AIMS) (2015). ISBN 978-1-4673-8675-3
16. Alzoabi, U.S., Alosaimi, N.M., Bedaiwi, A.S.: Parallelization of KMP string matching algorithm. In: World Congress on Computer and Information Technology (WCCIT). INSPEC Accession Number: 13826319 (2013)
17. Vance, A.: Hadoop, a free software program, finds uses beyond search, March 2009. <http://www.nytimes.com/2009/03/17/technology/business-computing/17cloud.html>
18. Vidanagama, D.: A comparative analysis of various string matching algorithms. In: International Research Conference, pp. 54–60 (2015)
19. Franek, F., Jennings, C.G., Smyth, W.F.: A simple fast hybrid pattern-matching algorithm. *J. Discrete Algorithms* **5**, 682–695 (2007)



# Robust Fingerprint Matching Based on Convolutional Neural Networks

Yanming Zhu, Xuefei Yin, and Jiankun Hu<sup>(✉)</sup>

School of Engineering and Information Technology, University of New South Wales,  
Canberra, ACT 2600, Australia

{yanming.zhu,xuefei.yin}@student.unsw.edu.au, J.Hu@adfa.edu.au

**Abstract.** Fingerprint has been widely used in biometric authentication systems due to its uniqueness and consistency. Despite tremendous progress made in automatic fingerprint identification systems (AFIS), highly efficient and accurate fingerprint matching remains a critical challenge. In this paper, we propose a novel fingerprint matching method based on Convolutional Neural Networks (ConvNets). The fingerprint matching problem is formulated as a classification system, in which an elaborately designed ConvNets is learned to classify each fingerprint pair as a match or not. A key contribution of this work is to directly learn relational features, which indicate identity similarities, from raw pixels of fingerprint pairs. In order to achieve robustness and characterize the similarities comprehensively, incomplete and partial fingerprint pairs were taken into account to extract complementary features. Experimental results on FVC2002 database demonstrate the high performance of the proposed method in terms of both false acceptance rate (FAR) and false rejection rate (FRR). Thanks to the robustness of feature extraction, the proposed method is applicable of incomplete and partial fingerprint matching.

**Keywords:** Fingerprint matching · Convolutional Neural Networks  
Fingerprint pairs · Relational features · Deep learning

## 1 Introduction

In recent years, biometric authentication has featured prominently for human verification and identification due to its robustness compared with password based security mechanism [6, 13, 18, 19, 22]. Among many biometrics, fingerprint has proved to be a very reliable human identification and verification index and has enjoyed superiority over other biometrics [14]. The dominance of fingerprint has been established by the continuous emergence of various automatic fingerprint identification systems (AFIS). A number of factors have caused bottlenecks towards achieving desired system performance, such as lack of reliable feature extraction algorithm, low accuracy of fingerprint alignment and difficulty of reliable similarity measurement between fingerprints [10, 12, 27]. Although lots of

efforts have been put into the development of a reliable system and tremendous progress has been made, we are still far from the goal.

Fingerprint matching, as the most popular and widely researched authentication system, can be classified into either identification or verification. Identification involves suggesting whether or not the query can find a match in the database, whereas verification involves deciding whether a query fingerprint matches the holding template. Both fingerprint identification and verification rely on accurate recognition of fingerprint features. The main challenges confronting fingerprint matching are the large intra-class variations (in fingerprint images from the same finger) and large inter-class similarity (between fingerprint images from different fingers) [26]. The intra-class variation can be caused by unequal pressure, partial overlap, non-linear distortion and sensor noise while the inter-class similarity is mainly due to limited number of fingerprint patterns [25].

Various methods have been developed to achieve effective fingerprint matching and a remarkable progress has been made. The fingerprint matching methods can be basically classified into two categories: (1) minutia-based methods [3, 5, 9, 15, 23] and (2) image-based methods [1, 17, 20, 24]. The minutia-based methods achieve efficient fingerprint matching by extracting more matching features besides minutia locations and orientations [5, 9, 23] or by constructing more complex structures [3, 15]. Jain et al. [9] proposed using pores and ridge contours besides minutia points and a three level matching strategy to achieve high-resolution fingerprint matching. Choi et al. [5] incorporated ridge feature with minutia and obtained good results. Thuy et al. [23] increased ridge-valley structure features and proposed the local Thin-Plate-Spline deformation model to deal with non-linear distorted fingerprints. Cappelli et al. [3] proposed a novel Minutia Cylinder-Code representation based on 3D cylinder structures to improve the matching effectiveness. Medina-Perez et al. [15] improved fingerprint verification by using Minutia triplets. These methods are suitable for one-to-many fingerprint matching and can gain favorable results. However, they are not compatible with different fingerprint sensors such as higher sensor resolution and large sensors. Moreover, minutia-based methods may lead to unsuccessful matching when the two fingerprint images have different number of minutiae points or when they possess the fingerprint portions without significant information. Therefore, image-based methods were proposed to solve the above problems.

The image-based methods consider the overall fingerprint characteristics rather than minutiae points alone and utilize more discriminatory information such as curvature, density and ridge thickness. Most of the image-based methods use filter bank to achieve the local texture feature analysis [14]. Sha et al. [20] proposed combining the directional features with average absolute deviation features to construct fingercode for the filter bank. Nanni and Lumini [17] proposed local binary pattern (LBP) based on a multi-resolution analysis of the local fingerprint patterns, which is a Gabor filter-based discriminator with low computational complexity. Tico et al. [24] proposed extracting local texture from the wavelet transform of a discrete image to achieve fingerprint recognition. Benhammadi et al. [1] proposed a hybrid descriptor to construct texture map by combining the minutiae orientations. However, these methods require

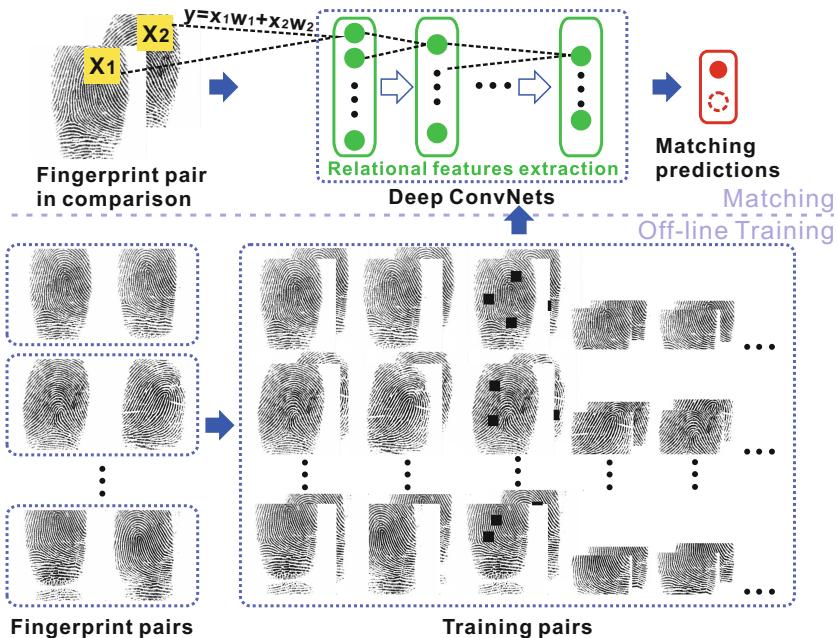
significantly larger storage space and higher running time and the performance is degraded on fingerprint images where minutiae points are not extract precisely and reliably.

Based on the above analysis and inspired by the superiority of convolutional neural networks for various classification tasks [2, 8, 11], in this paper, we proposed a novel fingerprint matching method based on Convolutional Neural Networks (ConvNets). The fingerprint matching problem is formulated as a classification system, in which an elaborately designed ConvNets is learned to classify each fingerprint pair as a match or not. A key contribution of this work is to directly and jointly learn relational features from raw pixels of fingerprint pairs. In order to achieve robustness and characterize the similarities comprehensively, incomplete and partial fingerprint pairs were utilized to extract complementary features and achieve data augmentation. By fully exploiting the spatial correlation between fingerprint pairs, the proposed method achieves high performance in terms of both FAR and FRR.

The rest of the paper is organized as follows: Sect. 2 presents the proposed methods in detail. Experimental results and analysis are provided in Sect. 3. The paper is concluded and the future work is discussed in Sect. 4.

## 2 The Proposed Method

In this section, we present the details of our method as depicted in Fig. 1. The basic idea of the proposed matching method is to classify each fingerprint pairs

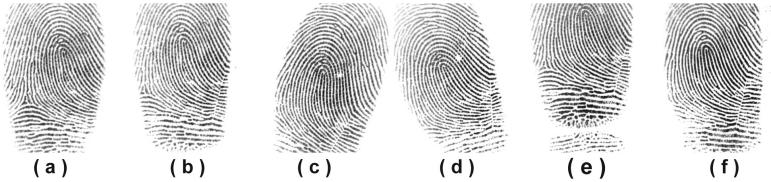


**Fig. 1.** Framework of the proposed method.

as either match or not using a elaborately designed ConvNets. For an input fingerprint pair in comparison, the relational features are extracted directly from the raw pixels by the pre-learned ConvNets (Sect. 2.2). Then, the extracted features are fully connected to a single neuron, which indicates whether or not the two impression images belong to the same fingerprint. In order to improve robustness of predictions, data augmentation is achieved by taking incomplete and partial fingerprint pairs into account (Sect. 2.1).

## 2.1 Fingerprint Pairs Preparation

Different from other fingerprint matching methods which extract features from each images separately and calculate the similarity at later stages, in this paper, we propose to jointly extract relational features from fingerprint image pairs. There are two kinds of fingerprint pairs: inter-class pairs constructed by fingerprints from different fingers and intra-class pairs constructed by fingerprints from the same finger. Due to the 2D acquisition characteristic of fingerprint image, the number of different impressions for one finger object is limited. Different impressions are usually varying at rotations and translations. A sample of different impressions of one finger object is shown in Fig. 2.



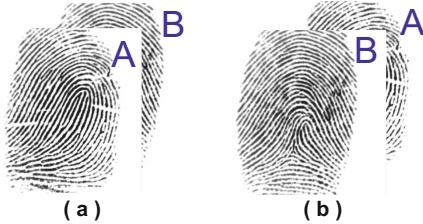
**Fig. 2.** Sample finger impressions from FVC2002 database: from (a) to (f) are the first six impressions of one finger used to construct the training pairs.

Based on this situation and on the purpose of improving robustness, we propose using incomplete and partial fingerprint image pairs to do the data augmentation. Figure 3 shows the proposed six variational fingerprint images based on one fingerprint impression. In order to extract relational features jointly, the fingerprint pairs are constructed by using fingerprint images with the same size. For example, in Fig. 3, fingerprint pair consists of image (a) and (b) is acceptable while fingerprint pair consists of image (a) and (c) is unacceptable. Considering the spatial correlation between fingerprint pairs, each input pair generates two modes by changing the order of two images. Figure 4 shows the two possible modes for a pair of fingerprint images.

In conclusion, for each finger object, 36 variational fingerprint images are generated by the proposed data augmentation method based on 6 original fingerprint impressions and are used to construct the intra-class pairs. Thanks to the proposed two modes, 396 intra-class pairs are finally generated to overcome the limitation of impression shortage for one finger object. In the experiment, we



**Fig. 3.** Variational fingerprint images for one fingerprint impression. From left to right: (a) the original fingerprint impression, (b) incomplete fingerprint image with random missing blocks, (c) and (d) partial fingerprint images sized  $\frac{4}{5}$  of original fingerprint image and (e) and (f) partial fingerprint images sized  $\frac{3}{5}$  of original fingerprint image. (c) and (d) and (e) and (f) differ slightly in the ranges of regions.



**Fig. 4.** Two modes for a pair of fingerprint images: (a) fingerprint pair mode with image A in the front and image B at the back and (b) fingerprint pair mode with image B in the front and image A at the back.

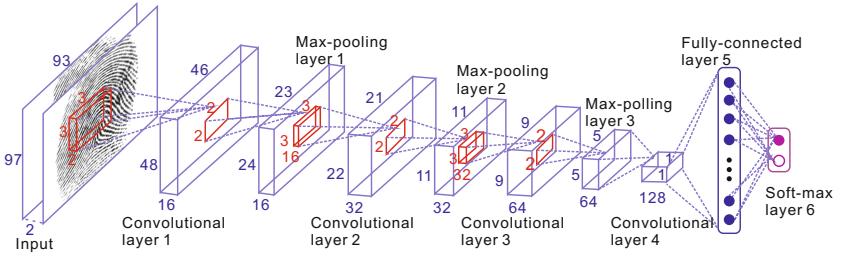
generate the same number of inter-class pairs by randomly selecting the impressions of different fingers.

## 2.2 ConvNets Architecture

The ConvNets is trained to classify a fingerprint pair in comparison as a match or not. The overall architecture of the proposed ConvNets model is shown in Fig. 5. The ConvNets consists of four convolutional layers, which can be expressed as:

$$y_j = b_j + \sum_i w_{ij} * x_i, \quad (1)$$

where  $*$  is the convolution,  $x_i$  and  $y_j$  are the  $i$ -th input and the  $j$ -th output, respectively.  $w_{ij}$  is the convolution kernel connecting the  $i$ -th input and the  $j$ -th output and  $b_j$  is the bias for the  $j$ -th output. According to the study in [21], successive convolution by small filters equals to one convolution operation by a larger filter but effectively enhances the model's discriminative power and reduces the number of filter parameters to learn. In this paper, we propose to downsize the input training pairs and use small filters of  $3 \times 3$  for all convolutional layers. For example, the original image size of  $388 \times 374$  in FVC2002 DB1 database is resized and downsized to  $97 \times 93$ . ReLU [7] activation function is utilized after all but the last convolutional layers, which can be express as:



**Fig. 5.** Sample finger impressions from FVC2002 database: from (a) to (f) are the first six impressions of one finger used to construct the training pairs.

$$y'_j = \max(0, y_j). \quad (2)$$

where  $y'_j$  is the output of  $y_j$  after ReLU. The dropout [11] is employed as a regularizer on the fully-connected layer in the case of overfitting. In the experiment, the dropout ratio is set to be 0.5. The proposed ConvNets output is a two-way softmax, which can be express as:

$$y_p = \frac{\exp(x_p)}{\sum_{q=1}^2 \exp(x_q)} \quad (p = 1, 2), \quad (3)$$

where  $x_p$  is the total input to an output neuron  $p$  and  $y_q$  is its output.

Since each layer extracts features from all the maps in the previous layer, relations between fingerprint pairs are modeled. With the network deepening, more global and higher-level relations between the two fingerprints are modeled. These high-level relational features make it possible for the top layer neurons in ConvNets to predict whether the two input fingerprint come from the same finger.

### 3 Experimental Results

In this section, we evaluate the performance of the proposed method on the most widely used public domain database FVC2002, which is an international competition database for fingerprint verification algorithms [16]. This database contains four distinct subsets: DB1, DB2, DB3 and DB4 and its summary is presented in Table 1. All the experiments are implemented in Matlab R2016a and run on a laptop with Intel Core i7 CPU at 2.6 GHz, 16 GB RAM and 256 GB hard drive.

Different from the existing fingerprint matching methods that apply similarity thresholding to predict the matching result of two fingerprints, the proposed method outputs a probability distribution over two classes (being the match or not), which directly indicates the matching result. Therefore, it is impossible to do the comparative experiments and to evaluate the performance of the proposed method in terms of some existing benchmark metrics such as ROC Curve and equal error rate (EER). In this paper, we use the following two metrics to quantitatively evaluate the performance of our method.

**Table 1.** Details of FVC2002 fingerprint database

	Sensor type	Image size (pixel)	Number	Resolution (dpi)
DB1	Optical sensor	$388 \times 374$ (142K)	$100 \times 8$	500
DB2	Optical sensor	$296 \times 560$ (162K)	$100 \times 8$	569
DB3	Capacitive sensor	$300 \times 300$ (88K)	$100 \times 8$	500
DB4	SFinGe V2.51	$288 \times 384$ (108K)	$100 \times 8$	About 500

**False Acceptance Rate (FAR).** This is the rate of occurrence of a scenario of two fingerprints from different fingers found to match, which is defined as:

$$FAR = \frac{N_f i}{N_t i + N_f i}, \quad (4)$$

where  $N_f i$  is the number of accepted imposter matches and  $N_t i$  is the number of rejected imposter matches.

**False Rejection Rate (FRR).** This is the rate of occurrence of a scenario of two fingerprint from same finger failing to match, which is defined as:

$$FRR = \frac{N_f g}{N_t g + N_f g}, \quad (5)$$

where  $N_f g$  is the number of rejected genuine matches and  $N_t g$  is the number of accepted genuine matches.

In the experiment, we use the first six impressions of each finger to train the ConvNets and use the remaining two impressions to do the test. As mentioned in Sect. 2.1, 396 intra-class pairs can be generated for each finger object. Therefore, we totally generate 39600 intra-class training pairs labeled as match and 39600 inter-class training pairs labeled as unmatch for each subset DB1, DB2, DB3 and DB4 to train the ConvNets, respectively. When the size of the input image changes in different subset, the map size in the following layers of the ConvNets will change accordingly. For each subset, each pair of two impressions are tested thus there are 100 genuine matches and 4950 imposter matches in our experiments. The fingerprint image pairs are firstly aligned by method in [4] to reduce the relative rotation between two fingerprint impressions and improve the robustness of relational feature extraction. Quantitative evaluation of the proposed method is shown in Table 2.

As shown in the table, FRR of 0% demonstrates the performance of the proposed method to model relational features and to identify fingerprint from the same finger. Meanwhile, the values of FAR are comparatively low, which makes the proposed method applicable in both commercial and criminal applications. The failure rates of matching for fingerprints from different fingers may be caused by the registration process, as both the rotation and translation operations tend to generate overlapping in fingerprints. Variations of the failure rates for different

**Table 2.** FAR and FRR vaules in percentage (%) for the four datasets

	FAR	FRR
DB1	1.69	0
DB2	1.15	0
DB3	0.42	0
DB4	0.79	0

subsets may be due to the differences in image quality and contrast of each subset. Visual inspection of fingerprints in the four subsets confirms this trend.

Thanks to the off-line training, the matching time of the proposed method is extremely short. The average matching time for one fingerprint pair is less than 1 s.

## 4 Conclusion

In this paper, we propose a novel fingerprint matching method based on Convolutional Neural Networks (ConvNets). The fingerprint matching problem is formulated as a classification system, in which an elaborately designed ConvNets is learned to classify each fingerprint pair as a match or not. A key contribution of this work is to directly and jointly learn relational features from raw pixels of fingerprint pairs. In order to achieve robustness and characterize the similarities comprehensively, incomplete and partial fingerprint pairs were utilized to extract complementary features and achieve data augmentation. By fully exploiting the spatial correlation between fingerprint pairs, the proposed method achieves high performance in terms of both FAR and FRR. Thanks to the robustness of relational feature extraction and extremely matching time, the proposed method is applicable to both commercial and criminal applications.

In the future, we will apply the proposed method to incomplete and partial fingerprint matching.

**Acknowledgments.** The authors would like to thank the support from ARC project LP120100595.

## References

1. Benhamadi, F., Amieouche, M.N., Hentous, H., Beghdad, K.B., Assanii, M.: Fingerprint matching from minutiae texture maps. *Pattern Recogn.* **40**(1), 189–197 (2007)
2. Cao, K., Jain, A.K.: Latent orientation field estimation via convolutional neural network. In: Proceedings of IEEE International Conference on Biometrics (ICB), pp. 349–356, September 2015
3. Cappelli, R., Ferrara, M., Maltoni, D.: Minutia cylinder-code: a new representation and matching technique for fingerprint recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**, 2128–2141 (2010)

4. Chang, S.H., Cheng, F.H., Hsu, W.H., Wu, G.Z.: Fast algorithm for point pattern-matching: invariant to translations, rotations and scale changes. *Pattern Recogn.* **30**, 311–320 (1997)
5. Choi, H., Choi, K., Kim, J.: Fingerprint matching incorporating ridge features with minutiae. *IEEE Trans. Inf. Forensics Secur.* **6**, 338–345 (2011)
6. Connie, T., Goh, M.K.O., Teoh, A.B.J.: A grassmannian approach to address view change problem in gait recognition. *IEEE Trans. Cybern.* **47**(6), 1395–1408 (2017)
7. Dahl, G.E., Sainath, T.N., Hinton, G.E.: Improving deep neural networks for LVCSR using rectified linear units and dropout. In: Proceedings of IEEE International Conference on Acoustic Speech Signal Process, pp. 8609–8613, May 2013
8. Ding, C., Tao, D.: Robust face recognition via multimodal deep face representation. *IEEE Trans. Multimed.* **17**(11), 2049–2058 (2015)
9. Jain, A.K., Chen, Y., Demirkus, M.: High-resolution fingerprint matching using level 3 features. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**, 15–27 (2007)
10. Jin, Z., Lim, M.H., Teoh, A.B.J., Goi, B.M., Tay, Y.H.: Generating fixed-length representation from minutiae using Kernel methods for fingerprint authentication. *IEEE Trans. Syst. Man Cybern.: Syst.* **46**(10), 1415–1428 (2016)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of Advance in Neural Information Processing Systems, pp. 1097–1105 (2012)
12. Lim, M.H., Teoh, A.B.J.: A novel encoding scheme for effective biometric discretization: linearly separable subcode. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(2), 300–313 (2013)
13. Low, C.Y., Teoh, A.B.J., Toh, K.A.: Stacking PCANet+: an overly simplified convnets baseline for face recognition. *IEEE Sig. Process. Lett.* **24**(11), 1581–1585 (2017)
14. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition, 2nd edn. Springer, London (2009). <https://doi.org/10.1007/978-1-84882-254-2>
15. Medina-Perez, M.A., Garcia-Borroto, M., Gutierrez-Rodriguez, A.E., Alramirano-Robles, L.: Improving fingerprint verification using minutiae triplets. *Sensors* **12**, 3148–3437 (2012)
16. Miao, D., Maltoni, D., Cappelli, R., Wayman, J.L., Jain, A.K.: FVC2002: second fingerprint verification competition. In: Proceedings of 16th International Conference on Pattern Recognition, pp. 811–814 (2002)
17. Nanni, L., Lumini, A.: Descriptors for image-based fingerprint matchers. *Expert Syst. Appl.* **36**(10), 12414–12422 (2009)
18. Oh, B.S., Jehyoung, J., Toh, K.A., Beng Jin Teoh, A., Jaihie, K.: A system for signature verification based on horizontal and vertical components in hand gestures. *IEEE Intell. Syst.* **28**(6), 52–55 (2013)
19. Oh, B.S., Toh, K.A., Teoh, A.B.J., Kim, J.: Combining local face image features for identity verification. *Neurocomputing* **74**(16), 2452–2463 (2011)
20. Sha, L., Zhao, F., Tang, X.: Improved fingercode for filterbank-based fingerprint matching. In: Proceedings of IEEE International Conference on Image Processing, pp. 895–898 (2003)
21. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). <http://arxiv.org/abs/1409.1556>
22. Sufi, F., Khalil, I., Hu, J.: ECG-based authentication. In: Stavroulakis, P., Stamp, M. (eds.) *Handbook of Information and Communication Security*, pp. 309–331. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-04117-4\\_17](https://doi.org/10.1007/978-3-642-04117-4_17)

23. Thuy, H., Thi, N., Huan, H.X., Ky, N.N.: An efficient method for fingerprint matching base on local point model. In: Proceedings of 2013 IEEE International Conference on Computing, Management and Telecommunications, pp. 334–339 (2013)
24. Tico, M., Immonen, E., Ramo, P., Kuosmanen, P., Saarinen, J.: Fingerprint recognition using wavelet features. In: Proceedings of IEEE International Symposium on Circuits and Systems, pp. 21–24 (2001)
25. Wang, Y., Hu, J.: Global ridge orientation modeling for partial fingerprint identification. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(1), 72–87 (2011)
26. Wang, Y., Hu, J., Phillips, D.: A fingerprint orientation model based on 2D fourier expansion (FOMFE) and its application to singular-point detection and fingerprint indexing. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(4), 573–585 (2007)
27. Wong, W.J., Teoh, A.B., Kho, Y.H., Wong, M.D.: Kernel PCA enabled bit-string representation for minutiae-based cancellable fingerprint template. *Pattern Recogn.* **51**, 197–208 (2016)



# A Personalized Multi-keyword Ranked Search Method Over Encrypted Cloud Data

Xue Tian<sup>1,2</sup>, Peisong Shen<sup>1,2</sup>, Tengfei Yang<sup>1,2</sup>, Chi Chen<sup>1,2</sup>, and Jiankun Hu<sup>3(✉)</sup>

<sup>1</sup> State Key Laboratory of Information Security, Institute of Information Engineering,  
CAS, Beijing 100093, China

{tianxue, shenpeisong, yangtengfei, chenchi}@iie.ac.cn

<sup>2</sup> School of Cyber Security, The University of Chinese Academy of Sciences,  
Beijing 100049, China

<sup>3</sup> Cyber Security Lab, School of Engineering and IT,

University of New South Wales at the Australian Defence Force Academy, Canberra, Australia  
J.Hu@adfa.edu.au

**Abstract.** Due to data privacy considerations, the data owners usually encrypt their documents before outsourcing to the cloud. The ability to search the encrypted documents is of great importance. Existing methods usually use the keywords to express users' query intention, however it's difficult for the users to construct a good query without the knowledge of document collection. This paper proposes a personalized ciphertext retrieval method based on relevance feedback, which utilizes user interaction to improve the correlation with the search results. The users only need to determine the relevance of the documents instead of constructing a good query, which can greatly improve the users query satisfaction. The selected IEEE published papers are taken as a sample of the experiment. The experimental results show that the proposed method is efficient and could raise the users' satisfaction. Compared with MRSE-HCI method, our method could achieve higher precision rate and equally high efficiency performance.

**Keywords:** Cloud computing · Ciphertext search · Multi-keyword search  
Relevance feedback

## 1 Introduction

Nowadays the data security issues in the cloud computing environment draws more and more attention. In order to ensure the privacy of personal data, users usually encrypt the documents before uploading them to the cloud server. However, data encryption makes the traditional retrieval mechanism invalid. Ciphertext search has become one of the important research issues in the field of information security.

A series of ciphertext retrieval methods have been proposed for different query requirements and application scenarios, including single keyword, conjunctive keyword and multi-keyword rank retrieval. However, the above methods are difficult to meet the needs of the users' personalized query. This is because these existing methods use the keywords to express users' query intention, which will encounter following three

challenges: (I) without knowledge of the targeted domain, it is difficult for the users to submit accurate keywords. (II) The same meaning can be expressed in different words, such as “aircraft” and “plane”. If a user queries the keyword “plane”, then the documents that contain the “aircraft” may not be returned. (III) Different users might have different occupations, interests or cultural backgrounds. Thus, even if they input the same query keywords, the focus of the search results will not be the same. In recent years, some methods such as semantic-based query [1], weighted keyword model [2] and PSRE (a personalized ciphertext search framework) [3] have been put forward to respond to challenge II or challenge III. However, the current query refinement methods all depend on the users to provide the accurate query keywords, which cannot be applied to cope with challenge I.

In order to tackle the above three challenges, we propose a personalized multi-keyword rank search method based on the similar search tree and relevance feedback (MRSE-SSF). The user query personalization is achieved through two-stage. The first stage allows users to submit query keywords and obtains the initial search results. In the second stage, the user picks out the relevant documents in the initial results and the optimized query keywords is automatically calculated and submitted. Then the server returns the final search results. The experiment shows that the method improves the users’ satisfaction about the search results.

The contribution of this paper is mainly reflected in the following three aspects:

- (1) A new type of ciphertext search method MRSE-SSF is proposed. By adopting the relevant feedback method, MRSE-SSF is used to enhance the user query satisfaction.
- (2) The similarity search tree structure is introduced in the ciphertext index building process to improve the query efficiency.
- (3) Experimental results demonstrate the effectiveness and efficiency of the proposed method.

## 2 Related Work

In recent years, many scholars contribute to the extensive study of the ciphertext retrieval. Song et al. [4] first described the ciphertext retrieval problem in 2000; a single-keyword query scheme based on symmetric encryption algorithm is realized by sequential query method. In 2004, Boneh et al. [5] first proposed a single-keyword search method based on the public key encryption (PEKS), which was originally used in encrypted e-mail systems. Nevertheless, it needs a safe channel shared by the data users and servers, which limits the practicality of the method. Later a formal security index structure: Z index is introduced [6]. The index model is realized by a pseudo random function and bloom filter, which is resilient against the adaptive chosen keyword attack. However, Z index does not provide ranking query mechanism. By adding the correlation score in the inverted table, the secure ranked search method [7] is proposed. During the query phase, the cloud server only needs to return the top k related documents matching the query conditions. This method could reduce the bandwidth consumption. However, the above work can only solve the problem of single-keyword ciphertext retrieval.

In order to express the user's query intention comprehensively, multiple keywords search method are proposed, including the conjunctive keyword retrieval [8] and multi-keyword rank search method (MRSE) [9]. The conjunctive keyword method demands the results contain all the query keywords. The MRSE method returns the top  $k$  most relevant documents through the implementation of secure  $k$  nearest neighbor algorithm. The query response time of the MRSE method grows with the increase of the document collection, which is difficult to adapt to the demand of the massive data in the cloud computing. Many articles [10–12] are devoted to improving the efficiency of MRSE. The MRSE-HCI method [11] introduces the dynamic hierarchical clustering algorithm to the MRSE, which has an excellent efficiency performance.

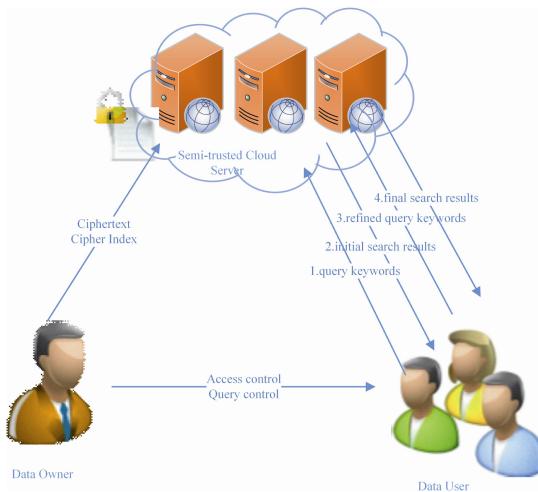
However, the methods above cannot solve the three challenges mentioned in sector 1, which includes inaccurate query keywords, synonyms in the query keywords and different focus of the search results. As to challenge II, some scholars proposed semantic-based query, which have implemented the query expansion of query keywords through the semantic dictionary (such as Wordnet) [1]. For the challenge III, the literature [2] allowed users to customize the keyword weight, in order to distinguish the query intention. [3] proposed a personalized ciphertext search framework PRSE. According to the user's search history, a scoring mechanism is established to express the user's personal search preferences. The existing query improvement methods rely on the user to provide accurate query keywords, which is not applicable in case of challenge I. To summarize, the current methods cannot solve all the above three challenges simultaneously.

### 3 Background and Related Definitions

This section describes the background of the ciphertext retrieval method. Section 3.1 gives the basic model of the personalized search method over encrypted cloud data. Section 3.2 describes the security model of the method. Section 3.3 presents the similarity search tree and the K-MEDOIDS algorithm used in the index building process. The evaluation criteria and main symbol definitions complete this section.

#### 3.1 System Model

The system model includes three entities: the data owner, the data user and the cloud server, as illustrated in Fig. 1. The data owner encrypts and outsources the massive document collection as well as the index built on them to the cloud server. In addition, the data owner entitles the data user to access the documents. Hence, the data user could search over the encrypted documents after authorization. (The authorization and access control are not discussed in this paper.) Meanwhile the cloud server provides the cloud storage and computing resources for the ciphertext retrieval. Once received the encrypted query request from the data user, the cloud server begins to search the index and returns the top  $k$  most relevant documents. So far, it is the classic system model of ciphertext retrieval. This paper introduces query refinement to the system model, as shown in step 3 and step 4 of Fig. 1. The user identifies the relevant documents to refine the keywords, and the cloud server returns the final search results.



**Fig. 1.** The system model of personalized ciphertext retrieval

### 3.2 Threat Model

This paper assumes that the data owner and the data user are trustworthy, while the cloud server is semi-trusted. That means the cloud server would perform the data user's request truthfully but might be curious about the documents' content. According to the information obtained by the cloud server, two threat models are discussed here [9].

*Known Ciphertext Model.* The cloud server is merely aware of the encrypted data in the cloud. That is, the encrypted documents, the encrypted index and the encrypted query vector.

*Known Background Model.* The cloud server not only knows all the information the first model mentioned, but also acquires the statistical information of the dataset such as the document/keyword frequency.

### 3.3 Preliminaries

In this paper, similar search tree is adopted as the index storage structure, and a clustering algorithm is used in the process of index building.

Similarity Search tree (SS tree) [13], which is the deformation of the R tree, is constructed from bottom to top. The upper node is covering all elements of the lower nodes of hyper sphere. Each node contains a center point and radius. If the node is a leaf node, the center is the document vector value; if the node is the intermediate node, the center is the hyper sphere's center.

The K-MEDOIDS algorithm [14] is a commonly used clustering algorithm. It mainly divides  $n$  data objects into  $K$  clusters. A medoid can be defined as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal. It can be viewed as the most centrally located point in the cluster.

### 3.4 Evaluation Standard

#### (1) Similarity Function

This paper calculates the dissimilarity rather than similarity between documents. Euclidean distance is adopted to measure the differences between pairs of documents. The documents' vectors such as  $x$  and  $y$  are first normalized. Then calculate the Euclidean distance of the vectors as Eq. (1).

$$\begin{aligned} D(x, y) &= \sqrt{|x|^2 - 2x \cdot y + |y|^2} \\ &= \sqrt{2 - 2x \cdot y} \end{aligned} \quad (1)$$

#### (2) Retrieval Result Evaluation

The users' satisfaction can be quantified by the search precision. The precision stands for the true correlation rate in the retrieval top  $k$  documents, which can be calculated as Eq. (2).

$$P = \frac{\text{the number of relevant documents in the results}}{\text{the number of retrieval documents}} \quad (2)$$

#### (3) Information Leakage Evaluation

This paper uses the same information leakage model as [9]. The top  $k$  documents are returned by the cloud server, and we try to leak retrieval order information to the cloud server as little as possible. The evaluation method of information leakage is explained below.

$$P_k = \sum_{i=1}^k |c_i - c'_i|/k \quad (3)$$

$c_i$  stands for the order of the file  $i$  in the top  $k$  retrieval.  $c'_i$  stands for the real position of file  $i$  without random value. The larger value of  $P_k$  is, the better the real retrieval results conceal and the less the information leaks.

### 3.5 Notations

Table 1 provides the notations in this paper.

**Table 1.** Notation.

$D$	The plaintext document collection, $D = \{d_1, d_2, \dots, d_m\}$ , where $m$ is the number of documents
$D_r$	The relevant plaintext document collection
$D_{nr}$	The irrelevant plaintext document collection
$C$	The encrypted document collection, $C = \{c_1, c_2, \dots, c_m\}$ , where $m$ is the number of documents
$W$	The dictionary. There are $n$ keywords, $W = \{w_1, w_2, \dots, w_n\}$
$D(O_i, O_j)$	The distance between the $O_i$ and $O_j$
$Q_w$	The query vector
$Q_{wopt}$	The optimized query vector
$R_{Qw}$	The radius of the query sphere
$R_n$	The radius of sphere in the similarity search tree
$min\text{-}SS$	The minimum number of members contained in the intermediate nodes of the SS tree
$max\text{-}SS$	The maximum number of members contained in the intermediate nodes of the SS tree
$u$	The number of the increased dimension to enhance the security of document vector
$v$	The number of dimension randomly chosen from the $u$ dimension

## 4 MRSE-SSF Method

The section describes the personalized multi-keyword ranked search method based on similarity search tree and relevance feedback (MRSE-SSF). In addition, the specific algorithms of each step are introduced separately.

### 4.1 MRSE-SSF Method

The MRSE-SSF method consists of five algorithms: (i) a key generation algorithm Keygen ( $I^n, u$ ), (ii) an encrypted index building method Build-index ( $D, SK$ ), (iii) a trapdoor producing algorithm Trapdoor ( $Q_w, SK$ ), (iv) a query request construction algorithm Query ( $T_w, k, I$ ) and (v) a query refinement algorithm Feedback ( $D_r, D_{nr}$ ).

- **Keygen ( $I^n, u$ ):** The data owner randomly generates a splitting indicator vector  $S$  which has  $(n + u + 1)$  bits and two reversible  $(n + u + 1) \times (n + u + 1)$  matrices  $M_1$  and  $M_2$ . And the symmetric key is defined as  $SK = \{S, M_1, M_2\}$ .
- **Build-index ( $D, SK$ ):** The data owner assigns a document vector  $D[i]$  to each document. The weight of keyword  $w_j$  in the document is recorded in  $D[i][j]$  ( $0 < j < n$ ) [15]. The index  $I$  is constructed according to the index building algorithm. The algorithm's details will be discussed in Sect. 4.3. Then we extend the center vectors in the index from  $n$  bits to  $(n + u + 1)$  bits. Particularly, the  $(n + j)$  ( $0 < j < u + 1$ ) bit is assigned to a random value  $R_j$  and the last  $(n + u + 1)$  bit equals 1. Next, based on

the splitting indicator vector  $S$ , the center vector is cut into two parts  $D[i][j]'$  and  $D[i][j]''$  and the index  $I = \{I', I''\}$ . If  $S_i$  is 1,  $D[i][j]'$  and  $D[i][j]''$  are random values, and the sum is  $D[i][j]$ ; if  $S_i$  is 0, then  $D[i][j]' = D[i][j]'' = D[i][j]$ . Last, the index  $I$  is encrypted by the matrices  $M_1$  and  $M_2$ , that is,  $I = \{M_1^T I', M_2^T I''\}$  and is outsourced to the cloud.

- **Trapdoor ( $Q_w, SK$ ):** The query vector  $Q_w$  has  $(n + u + 1)$  bits. The first  $n$ -bits value of  $Q_w$  is decided by whether the keyword occurred in the dictionary. If occurred, the corresponding position  $Q_w[i]$  is 1; if not, the position is 0. Then we randomly choose  $v$  from  $u$  keywords, and assign the relevant position to 1, and the else is 0. The last bit of  $Q_w$  is a random value  $t$ . The vector  $Q_w$  is then transformed by a random value  $q$ , that is  $Q_w = (q \cdot Q_w(n + u), t)$ . Afterwards  $Q_w$  is split according to vector  $S$ , and the process is reverse to the previous split. If  $S_i$  equals 1,  $Q_w[i]' = Q_w[i]'' = Q_w[i]$ ; if  $S_i$  equals 0,  $Q_w[i]'$  and  $Q_w[i]''$  are random values and the sum is  $Q_w[i]$ . Last, the matrices  $M_1$  and  $M_2$  encrypted  $Q_w'$  and  $Q_w''$ . The trapdoor is a triple,  $T_w = \{M_1^{-1} Q_w', M_2^{-1} Q_w'', R_{Q_w}\}$  and  $R_{Q_w}$  stands for the radius of the query sphere.
- **Query ( $T_w, k, I$ ):** During the query process, in order to improve the search efficiency, the cloud server performs the search hierarchically rather than iterating each document. On the basis of  $T_w$ , the cloud server firstly computes the relationship between the query sphere and the super sphere in the root note, and finds the sphere which has the max intersection with query sphere. Then search the next layer until the leaf nodes. Last, calculate the distance between the leaf nodes' document vector and the query sphere's document vector, and return the top  $k$  document list.
- **Feedback ( $D_r, D_{nr}$ ):** The data user divides the result set into the relevant document set  $D_r$  and the irrelevant document set  $D_{nr}$ , and calculates the optimized query keyword vector  $Q_{wopt}$ .

Then generate a new trapdoor  $T_w' = \text{Trapdoor}(Q_{wopt}, SK)$  based on the refined query keyword vector  $Q_{wopt}$ , and execute the query Query  $(T_w', k, I)$  again with the trapdoor  $T_w'$ .

## 4.2 MRSE-SSF Index Structure Construction Algorithm

In the second step of the MRSE-SSF method, the indexing structure is constructed. The similar search tree is introduced into the index structure, and the hierarchical structure of the document set is represented by the tree structure. The index construction is divided into the following steps:

Step 1: Set the minimum number of nodes  $m$  and the maximum number of nodes  $M$  in the similarity search tree.

Step 2: Call the K-MEDOIDS clustering algorithm and establish the  $N_0$  minimum sphere.

Step 3: Form a new sphere by uniting the  $x(m \leq x \leq M)$  spheres which are close. Call the K-MEDOIDS algorithm, and set  $k = 1$ . Then calculate the center point and the radius of the new sphere.

Step 4: Repeat step 3, choose the union result of which the increased volume is the minimum, output the  $N_i$  sphere.

Step 5: Set the abovementioned  $N_i$  sphere as one layer of similarity search tree.

Step 6: Repeat step 3, 4 and 5, until the number of sphere  $N_i$  less than  $m$ .

Step 7: The last  $N_i$  sphere is the root node of the similarity search tree.

### 4.3 Query Algorithm

The core idea of the query algorithm is to find the node in the index tree that has the largest intersection with the query hypersphere [13]. The positional relationship between the index tree hypersphere and the query hypersphere can be divided into three situations: contained, intersected and disjoint. Suppose that  $R_{Q_w}$  represents query hypersphere radius, which is set in the  $T_w$ , and  $R_n$  denotes hypersphere radius in the similarity search tree.  $O_{Q_w}$  stands for the center of the query vector that is the same as  $Q_w$ , and  $O_n$  represents the centroid value of the node in the similarity search tree.

The method of determining intersection, as shown in Eq. (4),

$$(R_{Q_w} + R_n) > D(O_{Q_w}, O_n) > |R_{Q_w} - R_n| \quad (4)$$

The method of determining containment, as shown in Eq. (5),

$$D(O_{Q_w}, O_n) < |R_{Q_w} - R_n| \quad (5)$$

The method of determining disjoint, as shown in Eq. (6).

$$D(O_{Q_w}, O_n) > (R_{Q_w} + R_n) \quad (6)$$

During the query process, we set the state flag to record the positional relationship for the above value. The state flag is in the range within four values: {unmarked, disjoint, intersected, contained}, and the initial value is unmarked.

The specific steps of the query algorithm are as follows:

Step 1: The server first calculates the relationship between the query sphere and the root nodes of each sphere in the index tree. Then get the maximum intersection between the query sphere with a certain sphere in root nodes.

Step 2: According to last step's result, continue to go down one layer of nodes to find the closest sphere to the query sphere.

Step 3: Repeat step 2, until go down to the layer of leaf nodes. Calculate the distance between the leaf nodes and the query sphere's center  $O_{Q_w}$  and choose the nearest node. Then return the top  $k$  nearest documents in that node to the user.

### 4.4 Relevance Feedback Algorithm

This paper adopts the Rocchio algorithm [16] as the feedback algorithm in the last step of the MRSE-SSF method. The algorithm is to implement the relevance feedback in the vector space model. The basic principle of the algorithm is to find an optimal query vector  $Q_{wopt}$ , which has the largest similarity between the relevant documents and the minimum degree of similarity between the unrelated documents. Use  $D_r$  to represent the

relevant document vector set, and denotes the irrelevant document vector set with  $D_{nr}$ , the optimal query vector  $Q_{wopt}$  can be expressed as:

$$Q_{wopt} = \arg \max_{Q_w} [\text{sim}(Q_w, \mu(D_r)) - \text{sim}(Q_w, \mu(D_{nr}))] \quad (7)$$

In the above formula,  $\mu(D)$  represents the center of the document vector set, which can be called a centroid, and is calculated as follows. The value of the centroid's each element is the mean of all the corresponding vectors' elements.

$$\mu(D) = \frac{1}{|D|} \sum_{\vec{d}_j \in D} \vec{d}_j \quad (8)$$

The optimal query vector is calculated as follows. The center of initial query vector is moved by a quantity that is the amount of difference between the centroid of the relevant document and the centroid of the irrelevant document.

$$\begin{aligned} \text{Feedback}(D_r, D_{nr}) &= Q_{wopt} \\ &= Q_w + [\mu(D_r) - \mu(D_{nr})] \\ &= Q_w + \left[ \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j \right] \end{aligned} \quad (9)$$

## 5 Security Analysis

### 5.1 Known Ciphertext Model

Under the known ciphertext model, the adversary can obtain the corresponding ciphertext information including the encrypted documents, the encrypted index and the encrypted query vector, but the encryption key is confidential. This method is based on the adaptation and security enhancement of the secure kNN method [17].

The encryption key of the method consists of two parts, which are the  $(n + u + 1)$  bits vector  $S$  and  $(n + u + 1) \times (n + u + 1)$  of the reversible matrix  $(M_1, M_2)$ . For simplicity and without loss of generality, we assume that  $u = 0$ . That is, do not add random values (add random values could increase the security of the method). Since the vector  $S$  is unknown, the document vectors  $D[i]'$  and  $D[i]''$  can be regarded as two random  $(n + 1)$  dimensional vectors. In order to solve the linear equations constructed by the encrypted document data, i.e.,  $C_i' = M_1^T D[i]'$  and  $C_i'' = M_2^T D[i]''$  ( $0 \leq i \leq m$ ), we have  $2m(n + 1)$  unknowns in  $m$  document vectors, and  $2(n + 1)(n + 1)$  unknowns in the matrix  $(M_1, M_2)$ . Because we merely have  $2m(n + 1)$  equations, less than the number of unknowns, there is not enough information to solve the document vector or matrix  $(M_1, M_2)$ .

Similarly, the query vectors  $Q_w$  and the optimized query vector  $Q_{wopt}$  can be inferred likewise. So the method is safe under the known ciphertext model.

## 5.2 Known Background Model

In this model, the cloud server not only knows the encrypted information, but also has the ability to analyze the query and results. Then we view the query irrelevance and keyword security in details.

**The Query Irrelevance.** Due to introducing the random value to the query vector generation process, the trapdoor is generated differently each time. Thus, the cloud server could not associate the query vector with the initial files. Three processes contribute to the randomness of the query vector.

1. Randomly choose  $v$  bits from the  $u$  bits and assign random values to the  $v$  bits.
2. Generate a random value  $q$  to normalize the  $(n + u)$  bits of the query vector.
3. The last bit of the query vector is set to the random value.

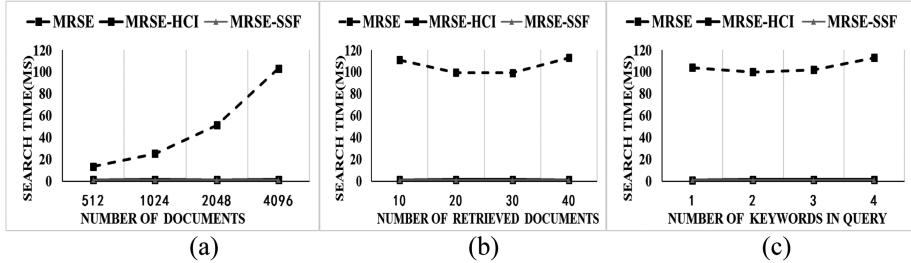
**The Keyword Security.** When data users are searching the documents, they tend to hide their query keywords from the cloud server. The trapdoors are usually used to protect the query keywords. In the known background model, the cloud server can infer the keywords by the word frequency information (including the number of documents including the keywords). In this paper, the keyword encryption method adopts the secure inner product calculation method [18, 19]. Because the encryption method is proved to meet the keyword security [18], the encryption method proposed in this paper conforms to the keyword security.

By setting the value of  $u$ , the user can confuse the relationship between the query and the search results, and increase the difficulty of using the statistical analysis to guess the keywords. However, the increase in the value of  $u$  will also reduce the accuracy of the query, therefore, this is the balance between the keyword security and the query accuracy.

## 6 Performance Analysis

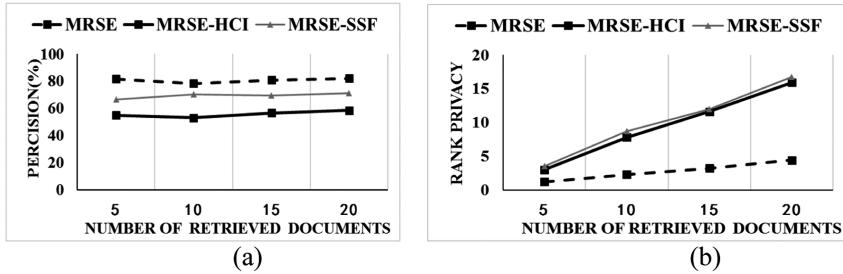
To test MRSE-SSF model's performance on the real dataset, we have established the experimental platform to verify the accuracy and efficiency of the retrieval. The testing platform is built on the Intel Core i7 3.40GZ Linux server, and the data set uses the selected papers published in the last 10 years of IEEE conference, which are 4196 documents and 8341 keywords. Figure 2 shows the efficiency of the MRSE-SSF method, the MRSE-HCI method and the MRSE method.

In Fig. 2(a), when the number of the document collection is exponentially increased, the query response time of MRSE method is exponential, and the query response time of MRSE-SSF and MRSE-HCI is linear. The query time of MRSE-SSF, MRSE and MRSE-HCI is stable in Fig. 2(b), when the number of retrieved documents varies. In Fig. 2(c), when the number of query keywords changes, the MRSE-SSF method still has a great advantage over MRSE.



**Fig. 2.** Search efficiency (a) search time for different number of documents (b) search time for different number of retrieved documents (c) search time for different number of keywords

Figure 3(a) shows the comparison of the MRSE-SSF, MRSE and MRSE-HCI methods in terms of precision. As can be seen from the picture, the precision of MRSE-SSF is improved compared to MRSE-HCI.



**Fig. 3.** (a) Precision (b) Rank privacy

Figure 3(b) shows the comparison of MRSE-SSF, MRSE and MRSE-HCI methods on the information leakage of the sorting results. The figure shows that the MRSE-SSF rank privacy is higher than MRSE, which means it protects privacy better.

In sum, the MRSE-SSF is more effective and has more privacy protection than the MRSE method, which has the similar advantages as the MRSE-HCI method. At the same time, the search results' precision of MRSE-SSF method is higher than the MRSE-HCI method, which is the focus of our paper's contribution.

## 7 Conclusions

In order to improve the quality of search results, this paper proposes a personalized multi-keyword ciphertext retrieval method based on relevance feedback. After the first stage of the query, the user feedbacks the interested documents in the results. Next, an optimized query vector is generated to further improve the search results. This method can provide the data users with personalized search results, which are more in line with the query intent of the users.

## References

1. Fu, Z., Sun, X., Ji, S., Xie, G.: Towards efficient content-aware search over encrypted outsourced data in cloud. In: IEEE INFOCOM 2016—the 35th Annual IEEE International Conference on Computer Communications, pp. 1–9. IEEE, April 2016
2. Li, H., Yang, Y., Luan, T.H., Liang, X., Zhou, L., Shen, X.S.: Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data. *IEEE Trans. Dependable Secure Comput.* **13**(3), 312–325 (2016)
3. Fu, Z., Ren, K., Shu, J., Sun, X., Huang, F.: Enabling personalized search over encrypted outsourced data with efficiency improvement. *IEEE Trans. Parallel Distrib. Syst.* **27**(9), 2546–2559 (2016)
4. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: Proceedings of 2000 IEEE Symposium on Security and Privacy, S&P 2000, pp. 44–55. IEEE (2000)
5. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) *EUROCRYPT 2004*. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-24676-3\\_30](https://doi.org/10.1007/978-3-540-24676-3_30)
6. Goh, E.J.: Secure indexes. *IACR Cryptology ePrint Archive* 2003, 216 (2003)
7. Wang, C., Cao, N., Li, J., Ren, K., Lou, W.: Secure ranked keyword search over encrypted cloud data. In: 2010 IEEE 30th International Conference on Distributed Computing Systems (ICDCS), pp. 253–262. IEEE, June 2010
8. Zhang, B., Zhang, F.: An efficient public key encryption with conjunctive-subset keywords search. *J. Netw. Comput. Appl.* **34**(1), 262–267 (2011)
9. Cao, N., Wang, C., Li, M., Ren, K., Lou, W.: Privacy-preserving multi-keyword ranked search over encrypted cloud data. In: Proceedings IEEE INFOCOM, pp. 829–837. IEEE Press, New York (2011)
10. Tian, X., Zhu, X., Shen, P., Chen, C., Zou, H.: Efficient ciphertext search method based on similarity search tree. *J. Softw.* **27**(6), 1566–1576 (2016). (in Chinese)
11. Chen, C., Zhu, X., Shen, P., Hu, J., Guo, S., Tari, Z., Zomaya, A.Y.: An efficient privacy-preserving ranked keyword search method. *IEEE Trans. Parallel Distrib. Syst.* **27**(4), 951–963 (2016)
12. Chen, C., Zhu, X., Shen, P., Hu, J.: A hierarchical clustering method for big data oriented ciphertext search. In: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 559–564. IEEE, April 2014
13. White, D.A., Jain, R.: Similarity indexing with the SS-tree. In: Proceedings of the Twelfth International Conference on Data Engineering, pp. 516–523. IEEE, February 1996
14. Kaufman, L., Rousseeuw, P.J.: *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 344. Wiley, Hoboken (2009)
15. Witten, I.H., Moffat, A., Bell, T.C.: *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, Burlington (1999)
16. Manning, C.D., Raghavan, P., Schütze, H.: *Introduction to Information Retrieval*, vol. 1, no. 1, p. 496. Cambridge University Press, Cambridge (2008)
17. Wong, W.K., Cheung, D.W.L., Kao, B., Mamoulis, N.: Secure kNN computation on encrypted databases. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, pp. 139–152. ACM, June 2009

18. Sun, W., Wang, B., Cao, N., Li, M., Lou, W., Hou, Y.T., Li, H.: Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. In: Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security, pp. 71–82. ACM, May 2013
19. Shen, P., Chen, C., Tian, X., Tian, J.: A similarity evaluation algorithm and its application in multi-keyword search on encrypted cloud data. In: Military Communications Conference, Milcom 2015, pp. 1218–1223. IEEE (2015)



# Application of Fuzzy Comprehensive Evaluation Method for Reservoir Well Logging Interpretation While Drilling

Zhaohua Zhou<sup>1</sup>, Shi Shi<sup>1</sup>, Shunan Ma<sup>2(✉)</sup>, and Jing Fu<sup>1</sup>

<sup>1</sup> PetroChina Research Institute of Petroleum Exploration and Development,  
Beijing 100083, China

{zhaohua69, shishi69}@petrochina.com.cn, fujing0602@126.com

<sup>2</sup> Institute of Information Engineering, Chinese Academy of Science,  
Beijing 100093, China  
mashunan@iie.ac.cn

**Abstract.** Reservoir classification and evaluation is the base for gas reservoir description. Well logging interpretation while drilling technique collects drilling logging signal in real-time through the sensor module, and transmits to the database server wirelessly. Well logging interpretation model is applied to reservoir information analysis, which is important to describe gas reservoirs accurately. Because of complicated geological conditions, there is a deviation in single well logging interpretation model. To solve the problem, a reservoir well logging evaluation while drilling method based on fuzzy comprehensive evaluation is proposed. Key parameters affecting reservoir evaluation, such as porosity, permeability and gas saturation are considered. Fully mining the information contained in GR, SP, AC and RT well logging data. Firstly, the reservoir is divided into gas, poor-gas, dry layer and water layer. For each well logging method, statistical method is used to calculate the subordinate intervals of each reservoir's parameters, and the membership degree is calculated to form the evaluation matrix of the well logging method. Then, the weight of each parameter is selected to form the comprehensive evaluation weight matrix, and fuzzy comprehensive evaluation result of well logging is computed. Finally, the comprehensive evaluation results of different well logging methods are composed to evaluation matrix, and fuzzy comprehensive evaluation method is used again to get the final reservoir evaluation category, so as to provide scientific basis for gas field development decision making.

**Keywords:** Fuzzy comprehensive evaluation · Well logging · Reservoir

## 1 Introduction

Well logging is an important oil field service work. It is important to quickly process and send the log data to the user at the oil field company on time. Oil and gas mines are mostly located in the swamps, deserts, basins and shallow sea area, where the transportation infrastructure is relatively backward and the construction environment is complex.

The application of mobile network technology in the oil and gas industry provides a powerful guarantee for the real-time control of oil and gas exploration and production process. Equipment detector transfers the collected production data back to the background system through wireless transmission equipment. In this way, each production and management department can grasp the working state of oil and gas well accurately and timely, which can improve production efficiency and provide new means for fine management of oil and gas well. Well logging interpretation while drilling collected drilling logging signal in real-time through the sensor module, and transmitted wirelessly to the database server. Then well logging interpretation information platform is applied to analyze the layer information of the current drilling rig, and the ground staff can monitor the results in real-time.

Well logging interpretation is an inversion problem. The parameters required for oil and gas geology can not be measured directly from the logs, but need to be gained by inversion of the interpretation model. Well logging interpretation should pay attention to the analysis of four properties of reservoir: electrical property, lithology, physical property, and hydrocarbon-bearing property.

Among the four properties of reservoir, hydrocarbon-bearing property is affected by the interaction of multiple geological factors. Such as, the gas volume is high, probably because of thin reservoir, low permeability, and difficult to mine. On the other hand, the gas volume is low, because of high permeability, thick gas, easy to mine. At present, the difficulty of well logging interpretation while drilling lies in the complicated geological conditions which leads to a large deviation in single well logging interpretation model. It is necessary to make use of several logging methods to improve accuracy of reservoir evaluation. In this paper, real-time data acquisition of mobile well logging while drilling is applied, and the comprehensive fuzzy evaluation method is applied to optimize the use of several logging methods, and a reservoir well logging evaluation while drilling method based on fuzzy comprehensive evaluation is proposed. Considering the parameters of each reservoir, especially the main parameters which affect the reservoir quality, the reservoir information contained in the well logging data is fully excavated, and the reservoir is comprehensively evaluated by the fuzzy mathematics method, which is more comprehensive and accurate to provide the basis for gas field development and favorable zone optimization.

## 2 Related Works

In the process of well logging interpretation, the parameters such as reservoir porosity, permeability and saturation are evaluated quantitatively. However, due to the technical conditions, the evaluation accuracy of these parameters is not high. In order to solve this difficulty, domestic and foreign scholars have proposed various methods to improve the accuracy of parameter calculation.

Ding et al. proposed the JM0D interpretation model by establishing the relationship between the capillary pressure and the J function of underground tight sandstone reservoir [1]. This method has been widely used in practical production, which has effectively improved the calculation accuracy of tight sandstone reservoir parameters, provided

more information about logging parameters, and played a vital role in the exploration and development of tight sandstone reservoir. Abu-Shanab et al. used the data of Nuclear Magnetic Resonance logging (NMR) and density logging (DEN) data in the calculation of porosity and other parameters, taking full account of the changes of lithology and pore fluid in tight sandstone reservoirs, and significantly improved the accuracy of porosity calculate, and provide good foundation for the calculation of other reservoir parameters [2].

Domestic scholars also made a lot of efforts in the research of tight sandstone reservoir parameters. Chai et al. used the BP neural network to calculate the parameters of tight sandstone reservoir [3]. Liu et al. changed the overall evaluation method of the tight sandstone, and evaluated the tight sandstone reservoir by system method, the important composition and the layer-by-layer method [4]. Yang et al. introduced the new evaluation parameters into the evaluation of the tight sandstone reservoir by experiment, and formed a new evaluation method [5]. Meng and Zhou proposed new methods to judge the dry layer, water layer and gas layer accurately according to accurate judgment of fluid parameters in the formation of Nuclear Magnetic Resonance logging [6, 7]. Wen et al. proposed that the pore parameters and pore throat parameters should not be considered only in the evaluation of tight sandstone reservoirs, and environmental factors should be added to improve the evaluation effect [8]. Wang used acoustic data on the reservoir fluid identification and quantitative evaluation, which shows unique advantages in actual processing [9]. Li used the support vector machine (SVM) algorithm based on particle swarm optimization to identify and predict the tight sandstone reservoir fluid, and establish a complete set of compact sandstone reservoir fluid logging evaluation software [10]. Zhuang used BP, wavelet and Elman neural networks to predict the production capacity of Sulige tight sandstone reservoir. The Elman neural network method with the best prediction effect was selected as the prediction model of fracturing productivity in Sulige region [11].

### 3 Selection of Reservoir Well Logging Comprehensive Evaluation Index

From variety of well logging information, reservoir well logging fuzzy comprehensive evaluation method is comprehensive evaluation of many reservoir parameters, including the qualitative and quantitative of the reservoir to find out their intrinsic relationship, so as to make classification of reservoirs and productivity estimation. The key factors affecting the reservoir are considered as much as possible. According to their geological factors to reflect the credibility and geological factors contribute to reservoir quality, the corresponding weights of these factors are given, and the formula is used to calculate the weights of various reservoir evaluation indexes. Finally, these indexes are integrated as the final index of reservoir evaluation.

The GR (Natural Gamma), SP (Spontaneous Potential), AC (Acoustic) and RT (Resistivity) well logging were selected based on the statistics of 755 wells, and the logging data were respectively normalized.

Based on the comprehensive analysis of reservoir evaluation indexes proposed by predecessors, three key evaluation indexes, porosity, permeability and gas saturation are selected according to the characteristics of comprehensive evaluation of gas reservoirs in Su 25 block.

Reservoir evaluation classification results: gas layer, poor-gas layer, dry layer and water layer.

Given the evaluation reservoir, the data objects that need to be processed are shown in Table 1.

**Table 1.** Well log information of evaluation reservoir

Well logging method	Porosity	Permeability	Gas saturation
<i>GR</i>	$\phi_G$	$K_G$	$S_G$
<i>SP</i>	$\phi_S$	$K_S$	$S_S$
<i>AC</i>	$\phi_A$	$K_A$	$S_A$
<i>RT</i>	$\phi_R$	$K_R$	$S_R$

The values of each parameter in Table 1 are the fuzzy values obtained by their corresponding logging curves.

According to the actual test result of perforation gas, the evaluation indexes of various well logging methods are analyzed. Using the statistical analysis method, we get the parameters' maximum and minimum values of each reservoir type in each logging method. For example, the statistical results of gas reservoir evaluation indexes are shown in Table 2.

**Table 2.** Statistical results of gas reservoir evaluation indexes

Gas reservoir evaluation indexes	Porosity		Permeability		Gas saturation	
	Max	Min	Max	Min	Max	Min
<i>GR</i>	$Max\ \phi_G$	$Min\ \phi_G$	$MaxK_G$	$MinK_G$	$MaxS_G$	$MinS_G$
<i>SP</i>	$Max\ \phi_S$	$Min\ \phi_S$	$MaxK_S$	$MinK_S$	$MaxS_S$	$MinS_S$
<i>AC</i>	$Max\ \phi_A$	$Min\ \phi_A$	$MaxK_A$	$MinK_A$	$MaxS_A$	$MinS_A$
<i>RT</i>	$Max\ \phi_R$	$Min\ \phi_R$	$MaxK_R$	$MinK_R$	$MaxS_R$	$MinS_R$

In the same way, the statistical evaluation indexes of other reservoir types can be calculated, such as water layer, dry layer and differential gas layer.

For the data of each well logging method, the membership function of each index X in the evaluation reservoir is constructed as follows.

$$\text{If } Min_i < X_i < Max_i, F(x) = \frac{X_i - Min_i}{Max_i - Min_i}; \\ \text{If } X_i \geq Max_i, F(x) = 1.0; \text{ else, } F(x) = 0.$$

According to the membership function, membership degree of the evaluation results of each well logging method is calculated. For example, the membership degree of each index for SP well logging is shown in Table 3.

**Table 3.** Index membership degree for SP well logging

Evaluation class	Porosity	Permeability	Gas saturation
Gas layer	0.7	0.8	0.1
Poor-gas layer	0.4	0.9	0.6
Dry layer	0.3	0.6	0.5
Water layer	0.5	0.2	0.8

The membership degree of each evaluation index is normalized:

$$X_{ij} = \frac{X_{ij}}{\sum_{i=0}^n X_{ij}};$$

The result is shown in Table 4.

**Table 4.** Normalized result of index membership degree for SP well logging

Evaluation class	Porosity	Permeability	Gas saturation
Gas layer	0.37	0.32	0.05
Poor-gas layer	0.21	0.36	0.3
Dry layer	0.16	0.24	0.25
Water layer	0.26	0.08	0.4

In the same way, the normalized results of index membership degree for other well logging methods can be calculated.

## 4 Well Logging Comprehensive Evaluation Method

Fuzzy comprehensive evaluation, which provides a high level of confidence in decision-making based on fuzzy logic, is a branch of artificial intelligence. It classifies or distinguishes things by means of analyzing fuzzy information as much as possible. By considering the various factors that influence a certain thing, fuzzy comprehensive evaluation method uses fuzzy mathematical methods and makes a scientific evaluation of its merits and shortcomings [12].

Fuzzy comprehensive evaluation uses some concepts in fuzzy mathematics to evaluate actual problems which are comprehensive and complex. Fuzzy comprehensive evaluation is one application of fuzzy mathematical method. The basic principle is as follows:

- (1) Identifying the factors that can be used to judge the target set and evaluation set;
- (2) Respectively determining the weights of the factors and their membership degree vector and obtaining a fuzzy evaluation matrix;
- (3) Operating the fuzzy evaluation matrix of factors and the fuzzy weight vector to normalize the result. The final result is the fuzzy evaluation result.

The method uses fuzzy mathematics theory, easy to understand and effective to judge complex problems, which can be applied to many fields [13]. In this paper, we use fuzzy comprehensive evaluation method to evaluate multiple well logging parameters reservoir.

Following is the algorithm procedure for completing the multiple well logging parameter reservoir evaluation:

- (1) The subordinate intervals of each evaluation index of each well logging method are obtained;
- (2) The membership degree of each evaluation index of each logging method is calculated to form the evaluation matrix  $R$ .
- (3) For each well logging method, the weight of each evaluation index is selected to form a weight matrix  $E$ , which can be computed by artificial intelligence methods such as neural networks.
- (4) According to step (2) and step (3), the fuzzy comprehensive evaluation is carried out to get the judgment results of each well logging method to the evaluation layer. Namely,  $B = E \circ R$ .
- (5) According to the effective sensitivity of each well logging method to the block, the index weight  $E_L$  of each logging method is chosen, and the weight of each logging method can be selected by classification statistics.
- (6) The judgment result  $B$  of each well logging method is used to form the well logging information evaluation matrix  $R_L = [B_0, B_1, B_2, B_3]$ .
- (7) According to step (5) and step (6), the fuzzy comprehensive evaluation is carried out again, to get the judgment results of various well logging methods on the evaluation layer. Namely,  $B_L = E_L \circ R_L$ .
- (8) Based on the principle of maximum subjection, the interval to which maximum subjection scale corresponds is selected  $b_i \in B_L$ . Namely, the evaluation reservoir is eventually evaluated as  $i$ th class.

## 5 Application Instance

Similar to Table 4, the normalized result of index membership degree for GR well logging is shown in Table 5.

**Table 5.** Normalized result of index membership degree for GR

Evaluation class	Porosity	Permeability	Gas saturation
Gas layer	0.27	0.23	0.18
Poor-gas layer	0.31	0.42	0.43
Dry layer	0.06	0.04	0.15
Water layer	0.36	0.31	0.24

Evaluation indexes weights of various logging methods are selected:

$$E_{4 \times 4} = \begin{bmatrix} 0.4 & 0.07 & 0.32 & 0.21 \\ 0.53 & 0.12 & 0.28 & 0.07 \\ 0.37 & 0.18 & 0.23 & 0.22 \\ 0.38 & 0.16 & 0.3 & 0.16 \end{bmatrix}$$

gas layer	poor- gas layer	dry layer	water layer
-----------	-----------------	-----------	-------------

$$R_L = E_i \circ R_i = \begin{bmatrix} 0.75 & 0.46 & 0.54 & 0.42 \\ 0.28 & 0.69 & 0.31 & 0.345 \\ 0.18 & 0.84 & 0.27 & 0.03 \\ 0.52 & 0.35 & 0.86 & 0.47 \end{bmatrix}$$

The weights are selected as:  $E_L = [0.13, 0.34, 0.38, 0.15]$ .

$B_L = E_L \circ R_L = [0.28, 0.38, 0.31, 0.34]$ . The maximum membership is 0.38, and the reservoir evaluation gained by the well logging information is poor-gas layer.

Note: In order to avoid the same final evaluation values, the calculation accuracy can be improved.

## 6 Conclusions

In this paper, a reservoir well logging evaluation while drilling method based on fuzzy comprehensive evaluation is proposed, and the real-time data acquisition of mobile well logging while drilling is applied. Key parameters affecting reservoir evaluation, such as porosity, permeability and gas saturation are considered. Firstly, the reservoir is divided into gas, poor-gas, dry layer and water layer. For each well logging method, statistical method is used to calculate the subordinate intervals of each reservoir's parameters, and the membership degree is calculated to form the evaluation matrix of the well logging method. Then, the weight of each parameter is selected to form the comprehensive evaluation weight matrix, and fuzzy comprehensive evaluation result of well logging is computed. Finally, the comprehensive evaluation results of different well logging methods are composed to evaluation matrix, and fuzzy comprehensive evaluation method is used again and the final reservoir evaluation category is gained.

**Acknowledgments.** This work has been supported by National Science and Technology Major Project (No. 2016ZX05047003).

## References

1. Ding, S., Pham, T., et al.: Integrated approach for reducing uncertainty in the estimation of formation water saturation and free water level in tight gas reservoirs-case studies. In: SCA International Symposium, pp. 1–12 (2002)
2. Abu-Shanab, M.M., Hamada, G.M., et al.: DMR technique improves tight gas porosity estimate. *Oil Gas J.* **104**(4), 12–13 (2005)
3. Chai, X.Y., Han, C., et al.: Log interpretation of deep thin tight reservoir with special lithology and high resistivity. *Well Logging Technol.* **23**(5), 350–354 (1999)
4. Liu, J.Y., Li, Y.J., Yu, R.T.: The development and application of the reservoir comprehensive and quantitative evaluation system. *Comput. Tech. Geophys. Geochem. Explor.* **26**(1), 33–36 (2004)
5. Yang, Z.M., Zhang, Y.Z., et al.: Comprehensive evaluation of reservoir in low-permeability oilfields. *Acta Petrolei Sin.* **27**(2), 64–67 (2006)
6. Meng, X.S., He, C.C., Guo, Y.F.: Using NMR logging to evaluate tight gas-bearing sandstone reservoir. *Well Logging Technol.* **27**(Suppl.), 1–4 (2003)
7. Zhou, S.X., Xu, Y.B., et al.: Prediction methods for physical properties of compacted argillaceous sandstone reservoir and its application. *Nat. Gas. Ind.* **24**(1), 40–43 (2004)
8. Wen, L., Liu, A.P., et al.: Method of evaluating upper Triassic tight sandstone reservoirs in west Sichuan Basin. *Nat. Gas. Ind.* **25**(Suppl.), 49–53 (2005)
9. Wang, F.: Research on the Application and Evaluation of Cross-Dipole Acoustic Logging to Interpretation of Tight Reservoirs, Jilin University (2013)
10. Li, D.: Research on Fluid Unit in Tight Sandstone Reservoirs of Sulige Gas Field, Jilin University (2014)
11. Zhuang, H.: Research on Gas Productivity Prediction Based on Logs for Post-frac Tight Sandstone Reservoirs in Sulige Area, Jilin University (2013)
12. Zhang, J.F., Deng, B.R.: Application of Fuzzy Mathematics. Geological Publishing House, Beijing (1991)
13. Du, D., Pang, Q., Wu, Y.: Modern Comprehensive Evaluation Methods and Case Selected. Tsinghua University Press, Beijing (2008)



# Factor Effects for Routing in a Delay-Tolerant Wireless Sensor Network for Lake Environment Monitoring

Rizza T. Loquias<sup>(✉)</sup>, Nestor Michael C. Tiglao, Jhoanna Rhodette I. Pedrasa, and Joel Joseph S. Marciano

Electrical and Electronics Engineering Institute,  
University of the Philippines Diliman, Quezon City, Philippines  
{rizza.loquias, nestor, joel}@eee.upd.edu.ph,  
jipedrasa@up.edu.ph

**Abstract.** Delay-tolerant wireless sensor networks (DTWSN) is a promising tool to facilitate communication in disruptive and challenged sensor network environments not usually catered by traditional systems. In this paper, DTWSN application to a real-life lake scenario is considered with the description of the routing problem and proposed solution. Opportunistic Network Environment (ONE) simulator was utilized to determine the performance of First Contact, Epidemic and Spray and Wait routing protocols on the map-based mobility model of the lake. Factors considered are the number of nodes, bit rate and ferry speed. Analyses of delivery probability, latency and overhead ratio as well as buffer time and hop count as metrics of performance evaluation against the protocols are done using JMP software. Results revealed that Spray and wait outperforms the other protocols for the given scenario.

**Keywords:** Wireless sensor networks · Delay tolerant networks  
DTN routing · ONE simulator · JMP software

## 1 Introduction

The challenges and required mechanisms for wireless sensor networks (WSN) [1, 2] had put forward a vast opportunity for innovation becoming evident in the market today with the increasing availability of smart sensor products for various deployments. Beyond its conventional uses, WSN deployment found its way in forests [3], inhospitable terrain such as volcanoes, lakes and remote places inaccessible for any wired service because of the limited or total absence of network infrastructures. The absence of a stable path and irregularity of radio propagation in this type of environment contributes to delays and loss of signal. Research activities are active in the development of delay-tolerant communication networks that will operate in this kind of environment [4] and the interoperability of such networks with the conventional TCP/IP network is provided by an overlay architecture known as delay-tolerant networking (DTN) described in RFC 4838 [13].

Sensor networks that have features of both WSN and DTN are termed as delay-tolerant wireless sensor networks (DTWSN). It is a network deployment of sensor nodes where there are disruptions in the network connectivity because connection paths among nodes suffer disconnections; there are relatively long and variable delays, encounters high losses in the communication link and high error rate. Some real-world applications of DTN to sensor networks are described in [8] that include wildlife tracking, village communication network [5], social-based mobile networks [6] and disaster response ad-hoc networks [7]. It was envisioned by the interest group, DTNRG, that delay tolerant networks R & D activities and implementations will soon provide communication services to undeveloped parts of the world where there is scarce communication facility/infrastructure. A survey of projects in DTN applied to sensor networks is found in DTN-The State of the Art published by N4C [8].

This paper provides the following contributions: (1) describe a scenario for a scheduled-opportunistic routing in a delay-tolerant wireless sensor network for Lake Environment monitoring, (2) perform simulations and analyze the performance evaluation of three routing protocols used in a delay tolerant network as applied to the lake scenario. The scope is limited to the performance evaluation only of three routing protocols used in a delay tolerant network applied to the mobility model of the cited scenario. The radio performance of the delay-tolerant WSN and hardware design is part of future work.

The paper is organized as follows: Sect. 2 discusses the motivating scenario and related work. Section 3 provides an overview of delay-tolerant wireless sensor networks and routing protocols as well as describes the routing in a lake environment. Section 4 describes the simulation and design of experiment used. Section 5 presents the results. Finally, Sect. 6 concludes the paper.

## 2 Motivating Scenario and Related Work

A lake seventeen square kilometers in area is considered as use-case for delay-tolerant WSN deployment for environment monitoring. Lake Buhi in Camarines Sur, Philippines is known to the world as the home of the world's smallest fish, the Sinarapan (*Mistichthys luzonensis*) or locally known in the area as the "tabyos" which is previously under the threat of near extinction caused by overfishing, low water quality and abusive use of the lake environment (Fig. 1). The local government has managed to issue ordinances for the protection of the endemic fish by designating a portion of the lake as a fish sanctuary and to mitigate other problems such as the recurring tilapia "fishkill" that results to loss of income thereby affecting the livelihood of the people in the area. Aside from implementing schemes for the management and biodiversity enhancement of the lake environment that includes removal of excessive fish cages and fish repopulation strategies, a policy framework [11] was also proposed to meet the need for regular, close monitoring of the water quality of the lake especially the fish sanctuary. Compared to the manual water quality detecting methods that takes a long time to gather data, deploying a monitoring system based on the concepts of DTWSN (delay-tolerant wireless sensor networks) would present significant advantages such as convenience in the monitoring



**Fig. 1.** Lake Buhi in Buhi, Camarines Sur, Philippines with Mt. Asog at the background (Left) and a typical motorized ferry boat (Right) that cruises its water.

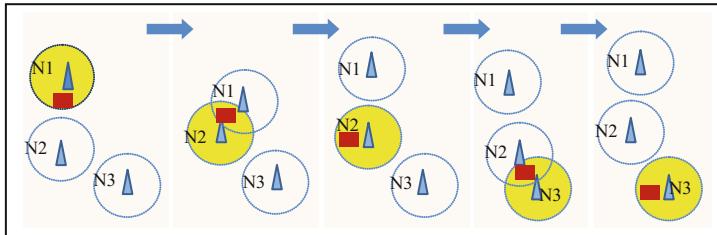
and faster collection of a variety of water parameters, a higher detection accuracy and enhanced data management of the monitoring system [9].

However, the design and set-up of the sensor network used in conventional indoor or short-range monitoring is not suited for remote large-area outdoor environment settings like forests and lakes because of the obvious difference in the landscape and circumstances. Relevant works [9, 10] developed hardware and software components of monitoring systems specifically addressing the lake environment and were consisting of data monitoring nodes/modules, data base station and remote monitoring center. Zigbee technology and GPRS/GSM modules were utilized to connect to the data server. The one used in Lake Palikpikan in Laguna made use of a novel sensor system with aerator that measures sensor data at two different depths over a period of one year. The project utilized UAV imaging over the lake to quantify fish cage density, water hyacinth coverage and disaster damage. It also utilized crowd-sourced participatory sensing by lake stakeholders through smartphone applications via cellular network. In this paper, delay-tolerant WSN for lake monitoring was explored using public ferry boats as DTN agents. This is similar in concept to the work in [12] which tackled ferry-assisted data-gathering. Since public ferry boats ideally travel on schedules and follow specific routes but may incur delays in actual travel time and may divert from usual routes, then DTN concepts can be utilized to collect data from the sensor nodes as the boats travel across the target coverage area and then route the data to a server. Mobile nodes may also be utilized rendering a combined scheduled-opportunistic approach for the target application.

### 3 Delay-Tolerant Wireless Sensor Networks and the Routing Problem

RFC 4838 and the Bundle protocol (RFC 5050) [13, 14] describe the DTN architecture, how it operates as an overlay above the transport layers of the networks it interconnects with, and the key service it offers. Between the application layer and the transport layer in the DTN protocol stack is another layer called the bundle layer. It implements the store-and-forward message switching mechanism. “Bundles” are application data that has been processed in the bundle layer and passed to the transport layer. Network

disconnections are overcome by the so-called custody transfers that provide the end-to-end reliability across the DTN. The said layer hides the disconnection and delay from the application layer [4]. The nodes in a DTN have the support for longer storage and custody transfers (see Fig. 2). These features grant the sensor nodes the ability to exploit scheduled, predicted and opportunistic connectivity. The system that has this ability can operate under intermittent connections.



**Fig. 2.** Custody transfer and the routing in DTN leverages on its built-in store-and-forward mechanism. Initially, message bundle (B) is stored at node N1 then when N1 is in contact with N2, the bundle will be forwarded to N2 which will have custody of the bundle until it makes contact with N3, the final destination of the message.

The forwarding scheme that employs node to node retransmission achieves end-to-end reliability, owing to DTN's built-in mechanism that prevents data loss and corruption. Researchers argued that the existing communication protocols independently developed for WSN and DTN may not be suitable for DTWSN because most of the existing WSN assumes always available data path and existing DTN designs on the other hand do not fully consider practical node energy, storage and computational capabilities [5].

In [15], an evaluation framework for DTN routing was proposed with emphasis on providing a trade-off between maximizing the delivery ratio and minimizing the overhead. It also discussed two broad categories of routing protocols under unicasting: routing without infrastructure assistance and routing with infrastructure assistance. The one considered in this study is classified under the latter, specifically routing scheme that uses mobile node relay where changes in movement play an important role in routing performance. A *mobility model* is therefore a requirement to imitate the movement pattern of the targeted real world applications in a relevant manner. The scope of the paper is limited to three routing protocols.

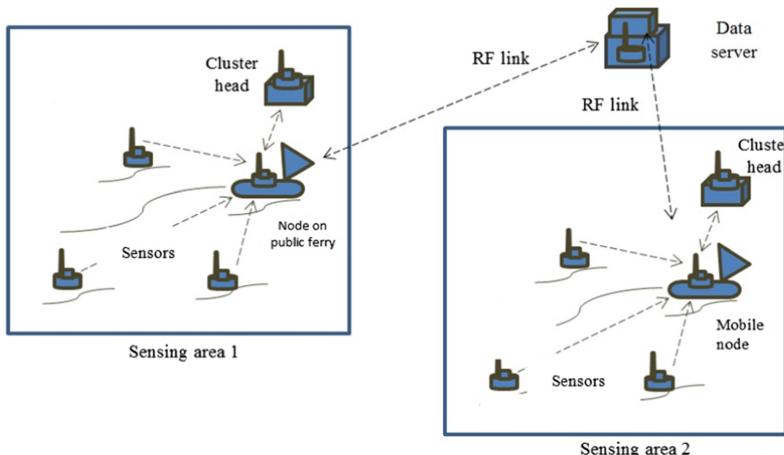
*First Contact* routing protocol dictates that the node forward messages to the first node it encounters along the way, this results in a “random walk” search for the destination node [15]. This is a technique where a node will transfer a single copy of the message to the first node it comes in contact with and it will continue until the message reaches the destination.

*Epidemic* routing [17], as one of the early proposed schemes that enable data delivery in intermittently connected mobile networks, is essentially a flooding protocol that replicates and propagates copies of a message to many mobile nodes within the network as well as retaining a copy of the message for a period of time. As its name

suggests, a node replicates a message and forwards it in an infective manner to a susceptible one once contact happens due to their movement.

*Spray and Wait* [18] routing protocol is an improvement of the epidemic routing by putting a maximum limit on the number of copies of message the source node generates. It has two phases: the Spray phase is where  $M$  copies of messages are forwarded to  $M$  distinct nodes; while Wait phase is when the nodes encountered are not the destination node then must wait until direct transmission to the destination is possible. Both epidemic and spray and wait protocols assume no knowledge of network topology and nodes mobility.

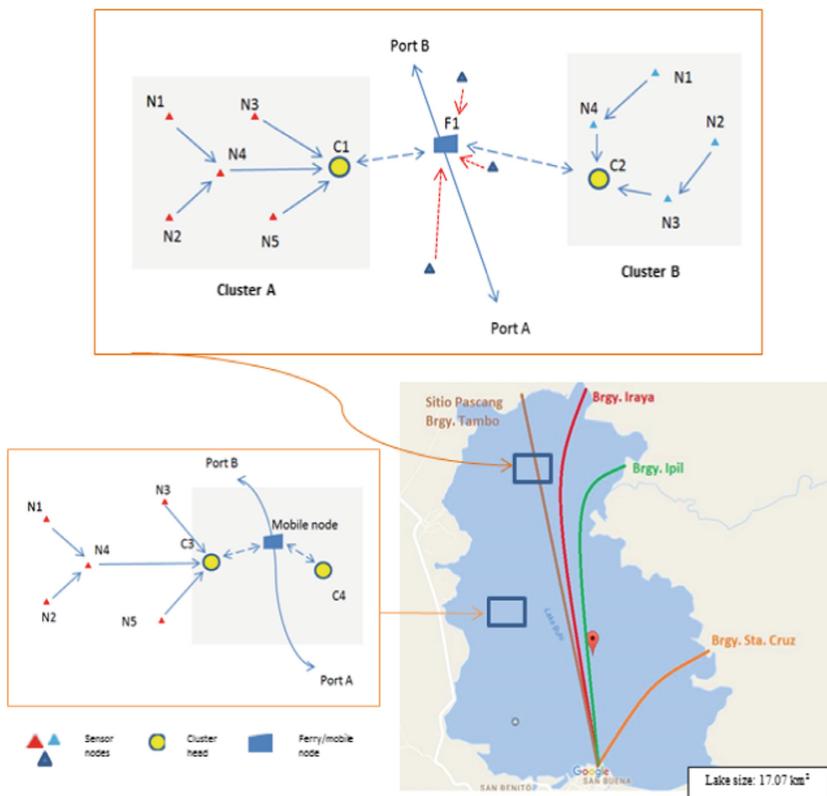
In a typical lake scenario, several public ferry boats traverse the water based on schedules and planned routes. In effect, the combined use of ferries and sensor nodes deployed in the water essentially make contact opportunities for data transfer. The simplified network model is shown in Fig. 3. There can be two groups of sensor nodes; one is clustered with designated cluster head while the other group consists of stand-alone nodes directly communicating to the ferry/mobile data collector. All are buoyed sensor nodes and are assumed to be fixed in position but may tolerate changes in location due to air and water movement. It is further assumed that the nodes are DTN-enabled meaning they are capable of store-and-forward routing.



**Fig. 3.** The simplified network model shows the basic elements. The sensor data are collected as the ferry and mobile nodes move along its route in a scheduled-opportunistic manner.

The mobile node (or message ferry) also serves as the DTN router cum network coordinator and cluster head (in cases where the cluster head is offline) and is capable of store-carry-forward routing. In the event that the functionality of these nodes is compromised, the node needs to be able to delegate the responsibility of ensuring data flow to another suitable node in the network. As a network coordinator/cluster head, it performs the wake-up call to sleeping nodes and performs data aggregations. As the mobile node moves in close proximity to the field sensors, data is transferred to the mobile node for later forwarding to the server.

At the start of operation, the server located at the ferry terminal will initiate request order for data. It will search for active routers (F1, Mobile node) to act as network coordinators by means of a status report. Active routers then broadcast the packet to awaken the cluster heads (C1, C2, C3, C4) as well as the sensor nodes (N1, N2,...) which are in sleep mode most of the time. The nodes will measure the water parameters then sends data to the cluster head which perform data aggregation, stores the aggregated data for a period of time before forwarding to the ferry or mobile node upon contact. The nodes are capable of custody transfers and cooperative relay so that data will be passed from the distant node to the node in close proximity to the ferry. The area to be monitored is divided into clusters based on the assumed sensing range of the nodes (ferry) traversing that route. The schedule of data collection is assumed to be coinciding with that of the ferry boat travel schedule, in this case twice in the morning and twice in the afternoon with two ferry boats per routes with interval of two hours each. For each cluster of stationary sensor nodes, a node relaying algorithm will minimize transmission delay while the waiting delay between mobile ferries will be taken care of by the ferry route algorithm that assumes direct interactions between ferries. This proposed algorithm is just described here, the details of which will be provided in the future (Fig. 4).



**Fig. 4.** Map of the ferry boat routes and the routing problem

## 4 Simulation and Design of Experiment

Simulation is essential in the study of WSN. There are a number of network simulators available for free downloads but there is one that is gaining popularity due to its support for DTN routing as well as mobility modeling and visualization. It is the Opportunistic Network Environment simulator or ONE simulator [16, 19] and for this study, version 1.4.1 was utilized. Previous works in [7, 12] used ONE simulator to simulate and analyze existing DTN Protocols.

### 4.1 Mobility Model

With the desire to emulate the movement pattern of the targeted real world applications and considering the application cited in this paper, a mobility model was derived for the lake monitoring sensor network. In the ONE simulator, map-based mobility model was selected to constrain the movement of the nodes to paths (routes) defined in the map data. The map data of the lake was obtained in WKT format using OpenJUMP [21], an open-source GIS program.

### 4.2 Performance Metrics

The parameters evaluated in the simulation are delivery probability, latency and overhead ratio. By using these metrics, the impact of the mobility model on the protocol performance was drawn from.

*Delivery probability* is the ratio of the number of messages that reaches the destination to the number of total messages generated and is an indicator of how reliable the network is in terms of message delivery.

*Latency* or delivery delay is the time it takes for a message to be delivered from the source to the destination. In a DTN system, a longer transmission delay is permissible but improving time of delivery will benefit the performance.

*Overhead ratio* is the number of messages replicated divided by the total messages in the network. The overhead ratio implies the use of network resources and buffer space due to the use of multiple copies of the same message to increase delivery chances.

Also considered in the results are *Hop count* which is the number of times the messages are exchanged between nodes before reaching the destination, and *Average buffer time* which is defined as the average time incurred by all messages that are delivered abandoned or stranded at the intermediate node buffers.

### 4.3 Factors

The factors considered in the simulation are: the number of nodes that vary from 12 to 40, data rate of the wireless interface used in the simulation is 802.11 or Wi-Fi that varies from 40 to 1375 kbps representing low-, medium-, to high-speed data rate, and ferry speed that typically varies from 0.5 to 3.5 m/s. The conduct of the experiments by network simulation had tried to model how these factors impact the performance of the protocol for the given scenario.

#### 4.4 Simulation Parameters

The parameters used in the simulation are listed in Table 1. The complete set-up of the simulation environment is listed in a text-based configuration file that contains the parameters. Data resulting from the simulations were retrieved in the MessageStatReport text file generated by ONE Simulator.

**Table 1.** ONE simulation parameters adjusted in the settings for each set of runs.

Parameter	Value
Message size	500K–1 MB
Buffer size	50 M
Number of nodes	varied
Area (m <sup>2</sup> )	3400 × 4500
Interface	802.11
Data rate	varied
Sensing range (m)	30–100
Ferry speed	varied
Protocol used	FirstContact, Epidemic, SprayAndWait

#### 4.5 Design of Experiment and Performance Evaluation Using JMP Software

To help in the selection of inputs with which to compute the output of the ONE simulator experiment, Space Filling design was utilized. This is a design of experiment technique suitable for computer simulations because of the deterministic nature of the model. A goal of designed experiments on such model is to find a simpler approach that adequately predicts the system behavior over limited ranges of the factors [20]. The Fast Flexible Filling method was chosen because it is the only method that can accommodate categorical factors and constraints on the design space. The categorical factor refers to the type of protocols used in the simulation. To maximize the use of the collected data and enable better interpretation, the fit model tool of the JMP software was utilized to analyze the data; and specifically using the ANOVA, parameter estimates and prediction profiler. These had provided the basis for the evaluation and comparison of the performances of the different protocols for the cited scenario.

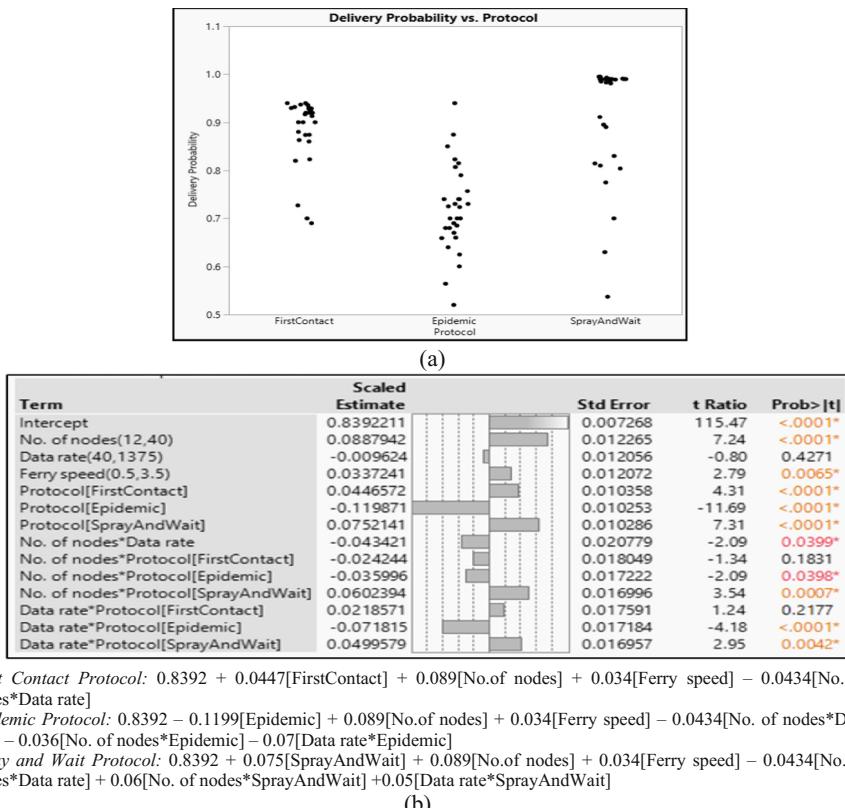
### 5 Results and Discussion

The effects of varying the factors: number of nodes, bit rate and ferry speed on delivery probability, latency, and overhead ratio were observed in the simulation and analyzed using the JMP software. The experiment consists of 30 runs for each protocol for a total of 90 runs. All the analyses were done using a level of significance of 0.05.

## 5.1 Factor Effects

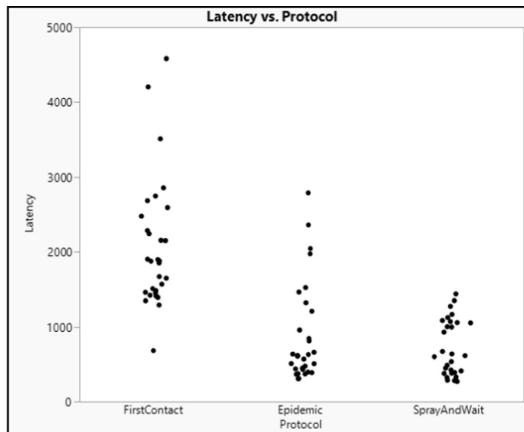
The data obtained from ONE simulation were inputted to JMP for analysis, the results of which are shown in the succeeding sections: the built-in graph builder provided the visual comparison on the protocol performance in terms of the given metrics and table of parameter estimates for the mathematical models of the responses for each of the protocols.

**Delivery Probability.** The fit model derived from the ANOVA revealed that the factor that significantly has effect on delivery probability is the number of nodes. This implies that as the number of nodes is increased, the reliability of message delivery is also improved due to the custody transfer mechanisms inherent to the nodes. However, increasing the number of nodes in the network would mean increased network cost. The results also revealed that delivery probability is also significantly affected by ferry speed and the interaction effects of number of nodes and data rate, and of number of nodes and type of protocol. For this metric, SprayAndWait performed well obtaining an almost 100% message delivery depicted in Fig. 5a and b.



**Fig. 5.** (a) Delivery probability vs. protocol (b) Parameter estimates using JMP

**Latency.** In terms of latency or delivery delay, the distribution plot shows a somewhat interesting pattern as shown in Fig. 6a. The values are dispersed in First Contact, less dispersed in Epidemic and least dispersed in SprayAndWait. The delay is almost reduced in half when SprayAndWait is used. This implies that the cooperation among the nodes in carrying the messages from other nodes speed up the delivery. Also with increasing data rate and ferry speed, the latency is reduced as revealed by the fit model in Fig. 6b.



(a)

Term	Scaled Estimate	Std Error	t Ratio	Prob> t
Intercept	1229.5359	48.72685	25.23	<.0001*
No. of nodes(12,40)	-709.3581	82.22511	-8.63	<.0001*
Data rate(40,1375)	-218.5721	80.82377	-2.70	0.0084*
Ferry speed(0.5,3.5)	-365.4387	80.93519	-4.52	<.0001*
Protocol[FirstContact]	882.41065	69.44147	12.71	<.0001*
Protocol[Epidemic]	-357.474	68.73961	-5.20	<.0001*
Protocol[SprayAndWait]	-524.9366	68.96117	-7.61	<.0001*
No. of nodes*Data rate	346.97069	139.3062	2.49	0.0148*
No. of nodes*Protocol[FirstContact]	-215.3028	121.0051	-1.78	0.0790
No. of nodes*Protocol[Epidemic]	-68.43108	115.4566	-0.59	0.5551
No. of nodes*Protocol[SprayAndWait]	283.7339	113.9424	2.49	0.0149*
Data rate*Protocol[FirstContact]	80.040637	117.9319	0.68	0.4993
Data rate*Protocol[Epidemic]	-35.54807	115.2051	-0.31	0.7585
Data rate*Protocol[SprayAndWait]	-44.49256	113.6841	-0.39	0.6966

*First Contact Protocol:*  $1229.5 + 882.4[\text{FirstContact}] - 709.36[\text{No.of nodes}] - 218.6[\text{Data Rate}] - 365.4[\text{Ferry speed}] + 346.97[\text{No.of nodes*Data rate}]$

*Epidemic Protocol:*  $1229.5 - 357.5[\text{Epidemic}] - 709.36[\text{No.of nodes}] - 218.6[\text{Data Rate}] - 365.4[\text{Ferry speed}] + 346.97[\text{No.of nodes*Data rate}]$

*Spray and Wait Protocol:*  $1229.5 - 524.9[\text{SprayAndWait}] - 709.36[\text{No.of nodes}] - 218.6[\text{Data Rate}] - 365.4[\text{Ferry speed}] + 346.97[\text{No.of nodes*Data rate}] + 283.7[\text{No.of nodes*SprayAndWait}]$

(b)

**Fig. 6.** (a) Latency (in ms) vs. protocol (b) Parameter estimates using JMP

From the analysis, latency is also significantly affected by ferry speed and the cross factor effects of number of nodes and type of protocol. Since the goal of the design is to minimize latency, therefore the protocol that performed well is the SprayAndWait

because it provided the fastest delivery of the messages by sending out multiple copies of the message. This however, will result to high buffer times as can be seen in Fig. 7.

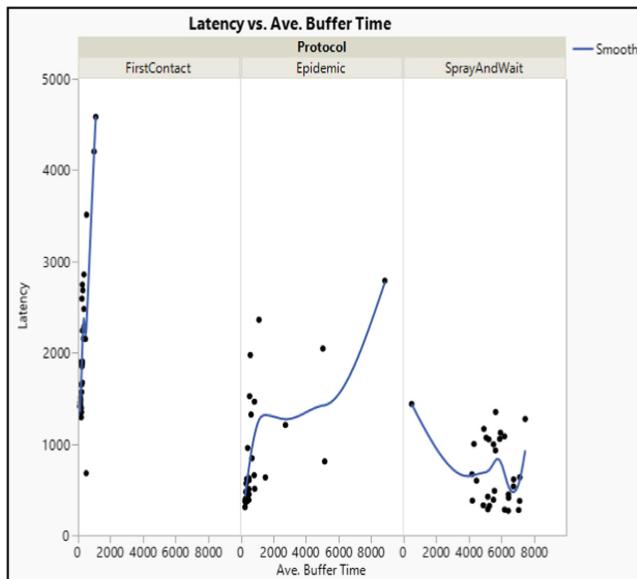
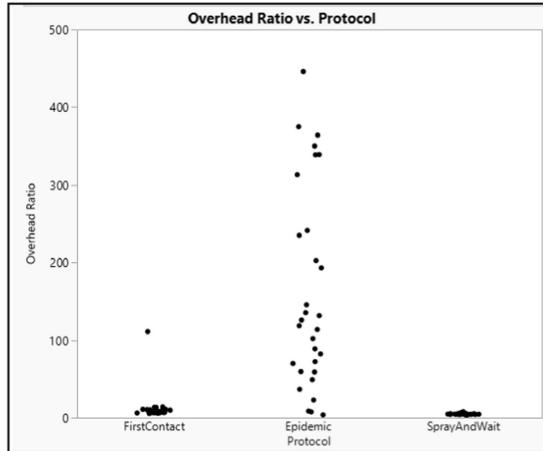


Fig. 7. Latency vs. average buffer time comparison using JMP

**Overhead Ratio.** As shown in Fig. 8a, the overhead ratio response for Epidemic protocol is dispersed from minimum to maximum while values for First Contact and SprayAndWait are closely intact at the minimum. This is expected since Epidemic has high overhead ratio because it makes more replications of messages than SprayAndWait. From Fig. 8b, the ferry speed has no effect on the overhead ratio.

**Average Buffer Time and Hop Count.** Interestingly in the results, SprayAndWait registered a higher average buffer time compared to First Contact and Epidemic. And as expected from the results, lower latency results to higher buffer times. This is not the time spent while in buffer but this is the time spent during transit between intermediate nodes. The performance of the routing protocols is influenced by the number of message copies they create, thus First contact being single-copy runs faster than Epidemic and SprayAndWait. In terms of hop counts, SprayAndWait utilized lesser hops than the other two protocols and we can deduce that it also consumes lesser energy because of the lesser number of hops required to deliver the message.

**Optimum Values.** The prediction profiler tool of the JMP software computed the desirable values for each of the protocols. Maximum desirability is provided by the SprayAndWait with the following values: 34 nodes, 530 kbps data rate, and 2 m/s ferry speed. However, these parameter values that will give the optimum routing



(a)

Term	Scaled Estimate	Std Error	t Ratio	Prob> t
Intercept	59.793716	3.017178	19.82	<.0001*
No. of nodes(12,40)	50.273469	5.091399	9.87	<.0001*
Data rate(40,1375)	42.320375	5.004628	8.46	<.0001*
Ferry speed(0.5,3.5)	4.0459548	5.011526	0.81	0.4219
Protocol[FirstContact]	-49.39945	4.299833	-11.49	<.0001*
Protocol[Epidemic]	102.38525	4.256373	24.05	<.0001*
Protocol[SprayAndWait]	-52.9858	4.270092	-12.41	<.0001*
No. of nodes*Data rate	46.356119	8.625872	5.37	<.0001*
No. of nodes*Protocol[FirstContact]	-43.22715	7.492664	-5.77	<.0001*
No. of nodes*Protocol[Epidemic]	92.067331	7.149101	12.88	<.0001*
No. of nodes*Protocol[SprayAndWait]	-48.84018	7.055338	-6.92	<.0001*
Data rate*Protocol[FirstContact]	-48.8092	7.302369	-6.68	<.0001*
Data rate*Protocol[Epidemic]	92.293079	7.133527	12.94	<.0001*
Data rate*Protocol[SprayAndWait]	-43.48388	7.03935	-6.18	<.0001*

First Contact Protocol:  $59.79 - 49.4[\text{FirstContact}] + 50.27[\text{No.of nodes}] + 42.3[\text{Data Rate}] + 46.36[\text{No. of nodes*Data rate}] - 43.23[\text{No. of nodes*FirstContact}] - 48.8[\text{Data Rate*FirstContact}]$   
 Epidemic Protocol:  $59.79 + 102.4[\text{Epidemic}] + 50.27[\text{No.of nodes}] + 42.3[\text{Data Rate}] + 46.36[\text{No. of nodes*Data rate}] + 92.06[\text{No. of nodes*Epidemic}] + 92.3[\text{Data Rate*Epidemic}]$   
 Spray and Wait Protocol:  $59.79 - 52.99[\text{SprayAndWait}] + 50.27[\text{No.of nodes}] + 42.3[\text{Data Rate}] + 46.36[\text{No. of nodes*Data rate}] - 48.8[\text{No. of nodes*SprayAndWait}] - 43.5[\text{Data Rate*SprayAndWait}]$

(b)

**Fig. 8.** (a) Overhead ratio vs. protocol (b) Parameter estimates using JMP

performance are considered in this paper as both theoretical and ideal. It is to be expected that practical results from testbed deployments will differ considering the actual cost and range.

## 5.2 Summary

The results obtained from the experiments and analysis showed that the increase in number of nodes has a slight effect on the delivery probability in Epidemic routing while using more nodes resulted to significant increase on the delivery probability for SprayAndWait. Epidemic has high overhead ratio since it make the most replications of

messages. This has consequences on storage capacity and energy consumption of the nodes. In terms of latency, SprayAndWait performed better than Epidemic but as the number of nodes was increased, both improved significantly while First Contact performed poorly. This implies that custody transfer and cooperation among the nodes speed up the message delivery. SprayAndWait utilized the least number of hops than epidemic and we can deduce that it also consumes lesser energy while First Contact utilized the most number of hops thus also utilizing the most energy. SprayAndWait registered higher average buffer time than the other two and it is expected because unlike Epidemic that performs flooding, SprayAndWait tends to “wait” until direct transmission to the destination is possible before transferring a message to a node. Buffer time in this context is not just the time spent while in buffer but added the time spent during transit between intermediate nodes. In over-all performance, SprayAndWait protocol is more favorable than Epidemic and First Contact. Maximum desirability is provided by the Spray and Wait protocol implying that this is the most suitable to the intended application. The results of the experiments validated the features of each of the protocols as described in the open literature.

## 6 Conclusion and Recommendation for Future Work

The evaluation of the protocol performance for the lake scenario considered the comparison of the effects of number of nodes, data rate, and ferry speed on delivery probability, latency, overhead ratio, average buffer time and hop counts. It was revealed by the results of the experiments that ferry speed has no significant effect on the protocol performance. However, this requires further investigation since it is a fact that mobility in a wireless radio system contributes to variations in the signal received. Map model of the lake scenario is utilized here to evaluate the three dominant DTN routing protocols. There are number of recently developed protocols that can be tested for this scenario. Energy expenditure which is an important design consideration needs to be tackled and incorporated to the proposed solution to the routing problem in a lake environment monitoring system under the premise of an intermittently connected delay tolerant network as described in this paper. Its full treatment can be part of future work.

## References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. *IEEE Commun. Mag.* **40**, 102–114 (2002)
2. Karl, H., Willig, A.: *Protocols and Architectures for Wireless Sensor Networks*. Wiley, Hoboken (2005)
3. Gay-Fernandez, J.A., Sanchez, M.G., Cuinas, I., Alejos, A.V.: Propagation analysis and deployment of a wireless sensor network in a forest. *Prog. Electromagnet. Res.* **106**, 121–145 (2010)
4. Fall, K.: A delay-tolerant network architecture for challenged internets. In: *Proceedings of SIGCOMM 2003*, pp. 27–34. ACM, New York (2003)
5. Li, Y., Radim, B.: A survey of protocols for intermittently connected delay-tolerant wireless sensor networks. *J. Netw. Comput. Appl.* **41**, 411–442 (2014)

6. Wei, K., Liang, X., Xu, K.: A survey of social-aware routing protocols in delay tolerant networks: Applications, taxonomy and design-related issues. *IEEE Commun. Surv. Tutor.* **16**(1), 556–578 (2014)
7. Chenji, H., Hassanzadeh, A., Won, M., Li, Y., Zhang, W., Yang, X., Stoleru, R., Zhou, G.: A Wireless Sensor, AdHoc and Delay Tolerant Network System for Disaster Response (2011). <https://engineering.tamu.edu/media/696805/2011-9-2.pdf>
8. N4C: Networking for communications challenged communities: architecture, test beds and innovative alliances (2007)
9. Del Rosario, J.M.: Deployment of a wireless sensor network for aquaculture and lake resource management. In: IEEE First International Workshop on Wireless Communication and Networking Technologies for Rural Enrichment (2011)
10. Solpico, D.B.: Towards a web-based decision system for Philippine lakes with UAV imaging, water quality wireless network sensing and stakeholder participation. In: ISSNIP 2015 (2015)
11. The Agroecosystems of Buhi: Problems and Opportunities. [http://pdf.usaid.gov/pdf\\_docs/Pnaau489.pdf](http://pdf.usaid.gov/pdf_docs/Pnaau489.pdf)
12. Alnuaimi, M., Shuaib, K., Alnuaimi, K., Abdel-Hafez, M.: Data gathering in delay tolerant wireless sensor networks using a ferry. *Sensors* **2015**(15), 25809–25830 (2015)
13. Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., Weiss, H.: Delay-tolerant networking architecture. RFC 4838 (Informational), April 2007
14. Scott, K., Burleigh, S.: Bundle protocol specification. RFC 5050 (Experimental), November 2007
15. Cao, Y., Sun, Z.: Routing in delay/disruption tolerant networks: a taxonomy, survey and challenges. *IEEE Commun. Surv. Tutor.* **15**(2), 654–677 (2013)
16. Keränen, A., Ott, J., Kärkkäinen, T.: The ONE simulator for DTN protocol evaluation. In: Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools 2009). ICST (Institute for Computer Sciences, Social-Informatics and Telecom Engineering), Brussels, Belgium, Article no. 55 (2009)
17. Vahdat, A., Becker, D.: Epidemic routing for partially connected ad hoc networks. Technical report CS-200006, Duke University, April 2000
18. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking, WDTN05, pp. 252–259. ACM, New York (2005)
19. The One. <https://www.netlab.tkk.fi/tutkimus/dtn/theone/>
20. Space-Filling Design. <http://www.jmp.com/support>
21. Open Jump. <http://www.openjump.org/>



# Estimating Public Opinion in Social Media Content Using Aspect-Based Opinion Mining

Yen Hong Tran<sup>1()</sup> and Quang Nhat Tran<sup>2</sup>

<sup>1</sup> People's Security Academy, Hanoi, Vietnam  
yenth.hvan@gmail.com

<sup>2</sup> University of New South Wales at ADFA, Canberra, Australia  
quang.tran@student.unsw.edu.au

**Abstract.** With the development of the Internet, social media has been the main platform for human to express opinions about products/services, key figures, socio-political and economic events... Besides the benefits that the platform offers, there are still various security threats relating to the fact that most extremist groups have been abusing social media to spread distorted beliefs, to incite the act of terrorism, politics, religions, to recruit, to raise funds and much more. These groups tend to include sentiment leading to illegal affairs such as terrorism, cyber-attacks, etc. when sharing their opinions and comments. Therefore, it is necessary to capture public opinions and social behaviors in social media content. This is a challenging research topic related to aspect-based opinion mining, which is the problem of determining what the exact opinions on specific aspects are rather than getting an overall positive or negative sentiment at the document level. For an entity, the main task is to detect all mentioned aspects of the entity and then produce a summary of each aspect's sentiment orientation. This paper proposes an aspect-based opinion mining model to address the problem of estimating public opinion in social media content. The model has two phases: 1 - extracting aspects based on double propagation techniques, and 2 - classifying opinions about the detected aspects with the consideration of the context of review sentences using the hybrid approach of machine learning and lexicon-based method.

**Keywords:** Aspect-based opinion mining · Aspect extraction  
Sentiment orientation · Public opinion analysis · Natural language processing  
Text mining · Social behavior

## 1 Introduction

With the proliferation of the Internet, massive user-generated content is posted in blogs, review sites, and especially social networks like Facebook, Twitter. The unprecedented volume as well as variety of user-generated content brings about new opportunities to understand social behavior and build socially-aware systems. This kind of data with subjective nature indicates public opinion. Public opinion influences and provides guidance for individuals, organizations, governments, and social communities during the decision-making process. While customer reviews might be useful for product sales and business, blogs and social networks can be used for political, religious, and security

issues. For example, messages in blogs that express social resentment at high intensity levels could be flagged as possible terrorist threats. Therefore, there exists an obligation to detect and categorize the opinions in social media to predict the user interest or behavior towards a specific domain, such as e-commerce, politics, security... This challenging task has foundations of natural language processing and text mining referred to opinion mining or sentiment analysis [1].

## 1.1 Opinion Mining

Khan et al. [2] states that an opinion represents the ideas, beliefs, and evaluations about a specific entity such as an event, a product, an organization or an individual. An opinion can be expressed in a variety of ways and generally has three main components: the source of the opinion (the opinion holder), the target of the opinion (the object about which opinion is expressed), and the opinion itself. It is simply a positive, negative or neutral view about an entity or an aspect of the entity from an opinion source. Positive, negative and neutral are called opinion orientation, sentiment orientation, semantic orientation or polarity. Opinion mining or sentiment analysis can be seen as the computational study of opinions, attitudes, and emotions toward entities and their different aspects [3]. Opinion mining has been an active research topic of knowledge discovery and data mining (KDD) in recent years due to its wide range of applications and many challenging research problems. Besides a variety of practical applications in commercial area such as summaries of customer's reviews, recommendation systems..., one of its potential application can be in political and security domain, such as internet public opinion monitoring and analyzing systems to help government intelligently understand, monitor sensitive public opinion and guide them [4]. Opinion mining can be used to examine social media networks to detect cyberbullying [5–7] or discussions concerning resentment society or planned criminals such as cyberattacks [8] with sophisticated attacker techniques and potential victims. Some recent research works have focused on applying opinion mining to detect security threats, such as terrorism [9, 10].

## 1.2 Aspect-Based Opinion Mining

Basically, there are three levels of opinion mining which have studied in the past decade (document level, sentence level and aspect level). Although opinion mining at document level and sentence level can be helpful in many cases, to obtain more fine-grained opinion analysis, it is necessary to delve into aspect level because positive (negative) evaluative text on an entity does not mean that the author has positive (negative) opinions on every its aspects. Aspect-based opinion mining provides opinions or sentiments about various aspects of a specific entity and entity itself. It was first called “feature-based opinion mining” in [11]. The basic task of aspect-based opinion mining is to extract aspects and summarize opinions expressed on aspects of entities. To mine opinion at aspect level, there are two core sub-tasks: 1 - extracting aspects of the entities in evaluative texts and 2 - determining sentiment polarities on aspects of entities.

The paper is organized as follows. In Sect. 2, we review and analyze some examples of previous work on aspect extraction and sentiment classification. We then describe

our proposed method for aspect extraction and sentiment analysis in Sect. 3. Section 4 contains evaluation of a case study in e-commerce domain. Finally, Sect. 5 draws conclusions and examines possibilities of future work.

## 2 Related Work

Hu and Liu [12] first proposed an unsupervised learning method based on association rules to extract product's aspects. The main idea of this technique is that users often use the same words for a specific aspect in their comments. Therefore, the frequent item sets which are nouns and noun phrases in the evaluative text are more likely to be the product's aspects. Input of Hu and Liu's aspect extraction model is a dataset of product's reviews. This dataset is transmitted to the extraction module after the preprocessing step (split sentences, part-of-speech tagging). The result obtained is a set of frequent aspects which are evaluatively mentioned by many reviewers ("frequent" means appearing in the dataset at a frequent rate greater than a determined experimental threshold). Based on this result, the system extracts evaluative words (opinion words) and detects infrequent aspects (with small number of occurrences). Aspect extraction method based on frequent item sets that Hu and Liu proposed requires a massive volume of reviews. However, extraction process still generates much noise, such as nouns or noun phrases which are frequent in both dataset and general language.

The method of Popescu and Etzinoni [13] is based on a similar idea of Hu and Liu [12]. However, their proposed technique can eliminate frequent phrases which are most likely not to be aspect expression based on the name of entity and Pointwise Mutual Information (PMI) score between the frequent phrases and the part-whole patterns like "of xx", "xx has", "xx comes with"..., in which "xx" is a word or phrase of entity. However, PMI copes with the problem of sparsity because bigrams composed of low-frequency words might receive a higher score than those composed of high frequency words. The extraction system also costs considerable time to incorporate the Web PMI statistics to review data in its assessment.

Qiu et al. [14] proposed double propagation algorithm. The idea of this approach is based on dependency relations between opinion words and aspect expressions. The opinion-aspect relationship is determined by a dependency parser. Knowing dependency relation and one of the two components (aspect expression or opinion word), the system can detect the remaining component. The extracted opinion words and aspects are then utilized to identify new opinion words and new aspects, which are used again to extract more opinion words and aspects. This process was repeated until no more opinion words or aspect expressions can be found. This algorithm is called double propagation because information spreads between opinion words and aspects after each iteration. Besides, this approach is also considered as a semi-supervised learning method because a small number of initial seeds are used to start the process of propagation. The effectiveness of this method depends on the selection of seeds at the initial step. In [14], initial seeds are randomly selected from an available list of opinion words. Thus, in the case, if there are no opinion seeds can be found from the evaluative text, the extraction will be ineffective. In addition, the propagation based on the syntactic rules is still generated much noise if

the size of dataset is large. This requires an effective method of noise removal to improve the accuracy.

In aspect-based opinion mining, after extracting aspect candidates from the evaluative dataset, the problem is to generate opinion summary for each aspect. However, users can use different words or phrases to mention one aspect, for example, “picture” and “image” are two different words but indicate the same aspect. Therefore, to create a meaningful summary, different expressions of one aspect should be grouped. There have been many methods proposed to solve this problem [15–17]. The key element of these learning algorithms is similarity score. There are two main approaches for similarity score, including: dictionary-based/lexical similarity and corpus-based/distributional similarity.

The other main task of opinion mining is sentiment orientation. Sentiment orientation is used to classify aspects, sentences or documents as positive, negative or neutral. Positive/negative polarity means that the opinion holder’s statement shows a positive/negative attitude toward the target object/aspect. Sentiment classification techniques can be divided into two categories: 1 - machine learning approach, and 2 - lexicon based approach [4]. Machine learning approach has the foundation of machine learning algorithms and linguistic features. The most frequently used algorithms for supervised sentiment classification are support vector machines (SVM), Naive Bayes classifier and Maximum entropy. Pang et al. [18] firstly adopted this approach to classify sentiment of movie reviews, however, they showed that the three machine learning methods they employed (Naive Bayes, maximum entropy classification, and support vector machines) do not perform as well on sentiment classification as on traditional topic-based categorization. The lexicon-based approach relies on a sentiment lexicon and is divided into dictionary-based approach [11] and corpus-based approach [19] which use statistical or semantic methods to find sentiment polarity. The dictionary-based approach finds opinion words, and then searches the dictionary of their synonyms and antonyms, therefore, it has a major disadvantage which is the inability to find opinion words in specific context domain. The corpus-based approach begins with a list of opinion seeds, and then finds other opinion words in a large corpus to solve the problem of context specific orientations. However, it is not a trivial task to prepare a such huge corpus.

### 3 Proposed Aspect-Based Opinion Mining Model

We use the term *entity* to denote the target object that has been evaluated. An entity can be represented as a tree and hierarchically decomposed based on the part-of relation. The root of the tree is the name of the entity. Each non-root node is a component or sub-component of the entity. Each link is a part-of relation. Each node is associated with a set of attributes. An opinion can be expressed on any node and any attribute of the node [3]. To simplify, we use the term *aspects* to denote both components and attributes.

Each entity  $E$  is represented with a finite set of aspects  $A = \{a_1, a_2 \dots a_n\}$  and each aspect  $a_i$  in  $A$  can be represented by a finite set of aspect expressions  $AE_i$ . A word or a phrase  $ae_{ik}$  ( $1 \leq k \leq |AE_i|$ ) in  $AE_i$  will be mentioned in a review sentence  $s_j$  and opinion orientation about aspect  $a_i$  in the sentence  $s_j$  will be expressed by using opinion

expressions  $oe_{ijh} \in OE_{ij}$ ,  $OE_{ij}$  is a finite set of opinion expressions in sentence  $s_j$  for aspect  $a_i$  ( $1 \leq h \leq |OE_{ij}|$ ). The objective of aspect-based opinion mining is to extract and group all phrases  $ae_{ik}$  in one aspect  $a_i$ , for each review sentence  $s_j$  discover all tuples  $(a_i, oe_{ijh}, s_j)$ , and finally generate an aggregated opinion summary for each aspect  $a_i$  through all review sentences  $s_j$ .

After studying some related research, we choose the extraction method based on the approach of Qiu et al. [14]. However, instead of semi-supervised learning with initial seeds of opinion words, we propose to use aspect seeds which are automatically selected from the input dataset with the orientation of a human-defined aspect sample. The human-defined aspect sample is domain-dependent and provided as the supplemental input of the system. To eliminate incorrect detected aspect candidates, the system has further steps that group aspect expressions  $ae_{ik}$  in each appropriate aspect node  $a_i$ . All tuples  $(a_i, oe_{ijh}, s_j)$  discovered from double propagation process will be assigned a sentiment orientation label using the hybrid approach of machine learning (Naïve Bayes classifier) and lexicon-based methods (Wordnet dictionary) with context consideration (dependency relations in each review sentences). An opinion summary for each aspect  $a_i$  of an entity  $E$  from input dataset will be generated as finally result.

Suppose that input dataset has been already collected and preprocessed, we propose an aspect-based opinion mining model with two phases: 1 - aspect and opinion word extraction, 2 - aspect-level sentiment classification and summary.

### 3.1 Aspect and Opinion Word Extraction

#### a. Generating aspect seeds

Generating aspect seeds is performed as follows: For a specific entity domain, there is a human-defined aspect sample playing role as the input of the module. Each phrase of this sample is split into individual word. The system searches for the appearances of these words in the input review text using simple string matching. The words appear to be aspects should be nouns. With “optical zoom”, a human-defined aspect in camera domain, for instance, the system searches for word “zoom” in the review texts and obtains noun phrases containing “zoom” as the potential aspect expressions (Fig. 1).

<b>body</b>	size weight design
<b>image</b>	image type resolution
<b>storage</b>	storage size storage type
<b>editing</b>	screen size viewfinder type display
<b>lens</b>	optical zoom digital zoom zoom range
<b>battery</b>	battery life
<b>sensor</b>	sensor type sensor size

**Fig. 1.** A human-defined aspect sample in camera domain

### b. Double Propagation

With the aspect seeds extracted previously, the system continues to expand the aspect set through the process of double propagation algorithm. Denote OA-Rel for the relationship between opinion words and aspects, OO-Rel for the relationship between opinion words and AA-Rel for the relationship between the aspects.

In double propagation algorithm [14] there are four sub-steps: (1) extract aspects using opinion words and OA-Rel relationship, (2) extract aspects using aspects and AA-Rel relationships, (3) extract opinion words using aspects and OA-Rel relationship, (4) extract opinion words using opinion words and OO-Rel relationship.

The input of the algorithm is aspect seeds A and evaluative dataset R. The processing steps in the algorithm are presented in detail in Fig. 2. The loop stops when not find any new aspects or opinion words. Here, we analyze an example to clarify the steps in the algorithm. Considering the following review:

*“Canon G3 gives great picture. The picture is amazing. You may have to get storage to store high quality pictures and recorded movies. And the software is amazing.”*

Suppose that the input of the algorithm has only one aspect as “picture”. In the first iteration, executing the command line 4 will extract opinion words “great” and “amazing”, then after the command line 5 executes we get “movies” as an aspect, performing the command line 11, we get aspect “software”. Finally, iterative process stops because there is no more any aspects or opinion words found. Thus, through the double propagation from an initial aspect seed, two other aspects and two opinion words detected.

---

Input: Aspect seeds A {aspectSeeds},  
           Evalutive dataset R  
 Output: Set of extracted aspects {aspectEx}  
           Set of extracted opinion words {opinionEx}

Algorithm: double propagation

```

1. {aspectEx} = {aspectSeeds};
2. {opinionStepi}=Ø; {aspectStepi}=Ø; {opinion}=Ø; {aspect}=Ø;
3. for each sentence s in R
4.     extract {opinionStepi} based on {aspectEx} using OA-Rel;
5.     extract {aspectStepi} based on {aspectEx} using AA-Rel;
6. endfor
7. set {opinionEx} = {opinionEx} + {opinionStepi};
8. set {aspectEx} = {aspectEx} + {aspectStepi};
9. for each sentence s in R
10.    extract {opinion} based on {opinionStepi} using OO-Rel;
11.    extract {aspect} based on {opinionStepi} using OA-Rel;
12. endfor
13. set {aspectStepi} = {aspectStepi} + {aspect};
14. set {opinionStepi} = {opinionStepi} + {opinion};
15. set {aspectEx} = {aspectEx} + {aspect};
16. set {opinionEx} = {opinionEx} + {opinion};
17. repeat 2 until (size{aspectStepi} = 0) and (size{opinionStepi} = 0);

```

---

**Fig. 2.** Double propagation algorithm

### c. Grouping aspects

Aspect candidates obtained from the previous steps are likely to contain a lot of redundancy. In this step, aspect grouping is to reduce redundancy and based on lexical similarity in WordNet [20] and a hierarchical structure of a human-defined aspect sample which is domain-dependent. Aspect grouping has some benefits: 1 - aspects are grouped into a hierarchical structure, for example, “weight” and “size” are grouped under father node “body”; 2 - reduce redundancy, for example, “picture”, “image”, “image quality” are grouped in one aspect node “image”.

The task of grouping aspects is equivalent to mapping each phrase in the set of extracted aspect candidates (AC) to a node in the human-defined aspect structure AT. The mapping process is performed based on phrase similarity metrics which are calculated from word similarity metrics.

- *Word similarity metrics*

Denote  $c_i$  and  $t_j$  are the corresponding phrases of the AC and AT, respectively

- Simple string matching

$$str\_match(c_i, t_j) = \begin{cases} 1 & \text{if } c_i \text{ matches } t_j \\ 0 & \text{if } c_i \text{ does not match } t_j \end{cases}$$

- Use information from WordNet and the type of word (part of speech).

In WordNet, each word is grouped into one or more synonymous sets called *synset* based on part-of-speech tags and semantics of the words. Each synset is a node in the taxonomy of the WordNet. If a word has more than one meaning, it will appear in many synsets at many positions in the taxonomy. Function  $\text{syns}(w)$  returns a set of all synsets that  $w$  belongs.

$$\text{syn\_score}(c_i, t_j) = \begin{cases} 1 & \text{if } \text{syns}(c_i) \cap \text{syns}(t_j) \neq \emptyset \\ 0 & \text{if } \text{syns}(c_i) \cap \text{syns}(t_j) = \emptyset \end{cases}$$

- Use some similarity measure  $sm$  introduced in [21]

Measuring semantic similarity between two synsets in the taxonomy WordNet has two approaches: the first is based on the distance between two nodes of the taxonomy corresponding to the two synsets, the second is relied on shared information of the two synsets which is the content of the nearest common parent node of them. Here, we use the second approach.

$$\text{sym\_score}_{sm}(c_i, t_j) = \frac{sm(c_i, t_j)}{\max(sm)}$$

$sm(c_i, t_j)$  can be calculated from one of the following expressions:

$$sm(c_i, t_j) = Res(c_i, t_j) = IC(LCS(c_i, t_j))$$

$$IC(c_i) = -\log Pr(c_i)$$

$$sm(c_i, t_j) = Lin(c_i, t_j) = \frac{2 \times Res(c_i, t_j)}{IC(c_i) + IC(t_j)}$$

$$sm(c_i, t_j) = Jcn(c_i, t_j) = \frac{1}{IC(c_i) + IC(t_j) - 2 \times Res(c_i, t_j)}$$

$IC(w)$  is the information content (*IC - Information Content*) of node  $w$  in WordNet.

$LCS(w_1, w_2)$  is the nearest common node (*LCS - Least Common Subsume*) of  $w_1$  and  $w_2$  in WordNet.

$Pr(w)$  is the probability of word  $w$  appear in the dictionary WordNet.

$Res(w_1, w_2)$ ,  $Lin(w_1, w_2)$ ,  $JCN(w_1, w_2)$  are the types of semantic similarity between  $w_1$  and  $w_2$ .

- *Phrase similarity metrics*

Denote  $ac_i$  and  $at_j$  are phrases in AC and AT respectively.  $c$  and  $t$  are the corresponding words in  $ac_i$  and  $at_j$ ;  $wm$  stands for similarity measure between words mentioned above.

- Function *max* returns the largest similarity measure between  $ac_i$  and  $at_j$

$$ac_i = \{c_1, \dots, c_n\}$$

$$at_j = \{t_1, \dots, t_m\}$$

$$\max(ac_i, at_j) = \max_{i,j} \{wm(c_i, t_j)\}$$

- Function *avg* returns the average similarity measure of  $ac_i$  and  $at_j$

$$ac_i = \{c_1, \dots, c_n\}$$

$$at_j = \{t_1, \dots, t_m\}$$

$$\text{avg}(ac_i, at_j) = \frac{\sum_{i=1}^n \max_j \{wm(c_i, t_j)\}}{n} + \frac{\sum_{j=1}^m \max_i \{wm(c_i, t_j)\}}{m}$$

$ac$  is mapped to  $at$  if  $ac$  and  $at$  has the highest similarity measure and this number is greater than a certain threshold  $\theta$ . With *str\_match* and *syn\_score*, threshold  $\theta = 0$ . With *sim\_score*, this threshold is set empirically.

#### d. Aspect-level sentiment orientation and summary

Given a set of sentiment orientation (SO) labels {positive, negative, neutral} and a set of tuples  $(a, o, s)$ , where  $o$  is a potential opinion word associated with aspect  $a$  in sentence  $s$ , the task is to assign an SO label to each tuple  $(a, o, s)$ . For example, the tuple *(image, poor, I am not happy with this poor image)* would be assigned a negative.

#### Find an SO label for each potential opinion word $o$

Assume that semantic orientation of word  $o$  is the class which maximizes the probability  $c$  conditional on  $o$ ,  $c \in C$  and  $C = \{\text{positive}, \text{negative}, \text{neutral}\}$ . Every word  $o$  can be represented as the set of its synonyms retrieved from WordNet.

$$\begin{aligned} SO(w) &= \operatorname{argmax}_{c \in C} P(c|o) \\ &= \operatorname{argmax}_{c \in C} P(o|c)P(c) \\ &= \operatorname{argmax}_{c \in C} P(\text{syn}_1, \text{syn}_2, \dots, \text{syn}_n|c)P(c) \\ &= \operatorname{argmax}_{c \in C} \frac{\sum_{i=1}^n \text{count}(\text{syn}_i, c)}{|\text{synset}_w|} P(c) \end{aligned}$$

$\text{syn}_1, \text{syn}_2, \dots, \text{syn}_n$  are synonyms of  $o$  and  $o$  is also considered as a synonym of itself.

For a synonym  $\text{syn}_i$ ,  $\text{count}(\text{syn}_i, c)$  is 1 if the synonym  $\text{syn}_i$  appears with polarity  $c$  in the dictionary of opinion lexicon [22], otherwise it is 0. Words that cannot be found in the opinion lexicon are assumed to have neutral polarity.

### Find an SO Label for Tuple $(a, o, s)$ Given the $o$ 's SO Label

First assign each tuple  $(a, o, s)$  an initial SO label which is  $o$ 's SO label. Then the system updates the default SO label whenever necessary based on syntactic relationships between opinion words and, respectively, between aspects. For example, (*memory, small, I hate the small memory because it shortly runs out of space.*) is a tuple detected. At the initial assignment,  $SO(\text{"small"}) = \text{"neural"}$ . However, in the context of sentence “I hate the small memory because it shortly runs out of space.”, “hate” and “small” satisfy modified rule and therefore it is expected that two these words have similar SO labels. Because “hate” is strongly negative, “small” in the context (memory, small, I hate the small memory because it shortly runs out of space.) acquires a negative SO label. To correctly update SO labels, the presence of negation modifiers is taken into consideration. For example, in the sentence “I don't like larger size because it is not convenient to handle”, the positive SO label of “like” is replaced with the negative labeled and then “large” in the context of the tuple (large, size, “I don't like larger size because it is not convenient to handle”) is inferred to have a negative SO label for aspect “size”.

The final aspect-level sentiment of an aspect  $a_i$  in sentence  $s_j$  is determined by a simple aggregation function which sums up the semantic orientation of all opinion words  $oe_{ijh}$  from all previously detected tuples  $(a_i, oe_{ijh}, s_j)$ . It is intuitive that an opinion phrase associated with an aspect will occur in its vicinity. Every semantic orientation is weighted relative to its distance to the aspect. The distance of the current opinion word and the aspect is the number of words lying in between. The idea behind this function is that opinion words which are closer to the aspect are most likely to be related to it.

$score(a_i, s_j) > 0$  means that a sentiment about the aspect  $a_i$  in sentence  $s_j$  is positive,  $score(a_i, s_j) < 0$  means that a sentiment about the aspect  $a_i$  in sentence  $s_j$  is negative,  $score(a_i, s_j) = 0$  means that a sentiment about the aspect  $a_i$  in sentence  $s_j$  is neural [23].

$$score(a_i, s_j) = \sum_{oe_{ijh} \in s_j} \frac{SO(a_i, oe_{ijh}, s_j)}{dist(oe_{ijh}, a_i)}$$

After all the previous steps, we are simply straightforward to generate the final aspect-level review summary. For each discovered aspect of the considered entity, each sentence in the input dataset which mentions this aspect is put into positive/negative/neural classes depending on the value of  $score(a_i, s_j)$ . A counter is computed to show how many review sentences give positive/negative/neural opinions about this aspect.

## 4 Case Study

This case study examines the performance of the proposed method for the problem of estimating sentiment of online camera reviews. This set of evaluative texts is collected from the site <http://epinions.com>, including 347 review posts for 8 types of cameras. Reviews for the same camera are stored in the same folder (Table 1).

**Table 1.** Experimental dataset of eight camera types

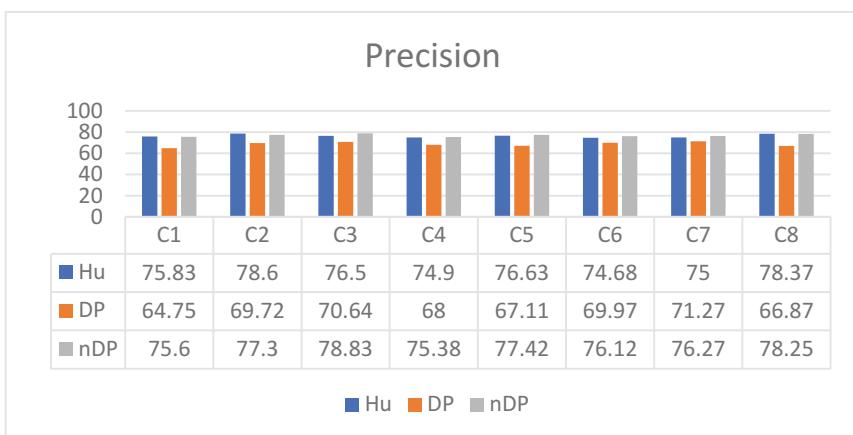
Type	Camera name	#Post	#Sentences
C1	Canon EOS 400D	65	953
C2	Canon Power Shot A510	44	714
C3	Canon Power Shot G3	45	593
C4	Canon Power Shot S100	50	286
C5	Nikon Coolpix 4300	34	358
C6	Nikon Coolpix L6	75	1591
C7	Panasonic Lumix DMC-FX7	20	684
C8	Sony Cyber-shot DSC-H1	14	307

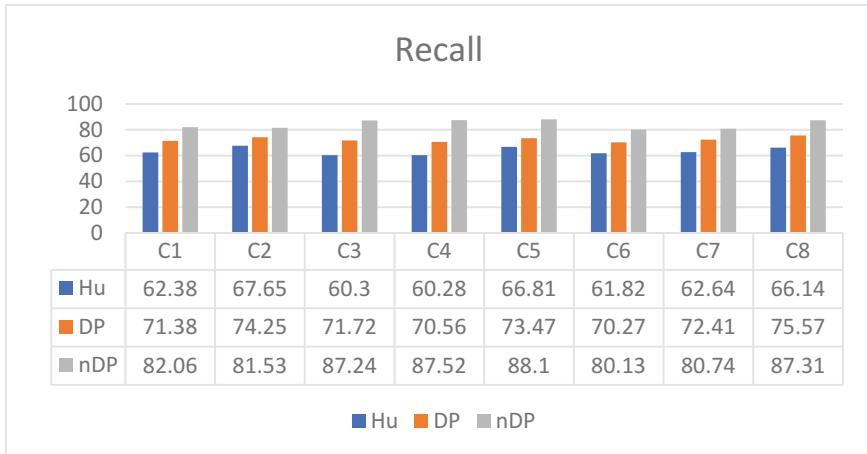
After processing review texts (sentences splitting, tokenizing, part of speech tagging, dependency parsing) using Html Agility Pack [24] and Stanford CoreNLP [25] we obtain linguistic information of each of 5486 sentences in dataset R. The input of system includes a human-defined aspect sample in camera domain, and the dataset R. After the phase of aspect extraction, we get a list of extracted aspects which are mapped (grouped) into appropriate aspects in the human-defined aspect sample using  $sym\_score$ ,  $avg$ ,  $Jcn$ ,  $\theta = 0.5$ ; and a set of tuples  $(a, o, s)$  where  $o$  is a potential opinion word associated with aspect  $a$  in sentence  $s$  (Figs. 3 and 4).

$$Precision = \frac{\#Correct\_Extracted\_Aspects}{\#Extracted\_Aspects}$$

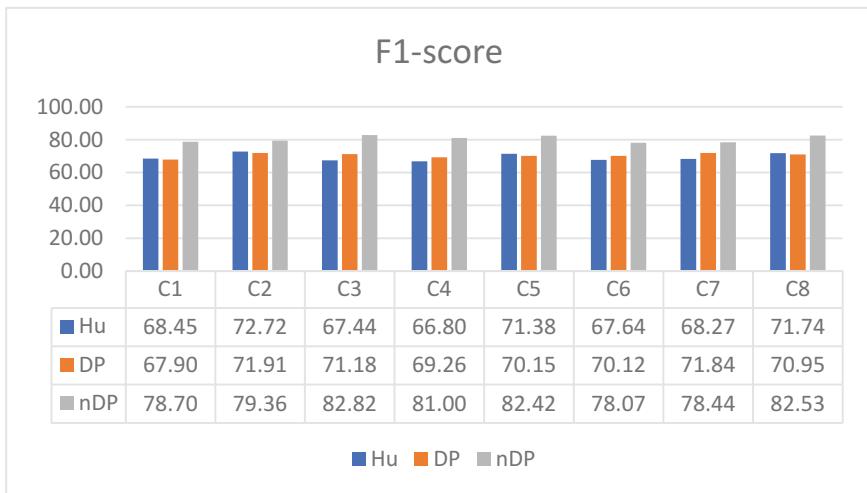
$$Recall = \frac{\#Correct\_Extracted\_Aspects}{\#Total\_Correct\_Aspects}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

**Fig. 3.** Precision

**Fig. 4.** Recall

See Fig. 5.

**Fig. 5.** F1

The charts above show the experimental results of our proposed extraction methods (nDP) compared to methods proposed by Qiu et al. [14] (DP) and methods based on the association rules of Hu and Liu [12] (Hu) in terms of precision, recall and F1 score. As can be seen from the three above charts, the precision of the proposed method (average 76.8%) is equivalent to that of Hu (average 76.3%), and both are higher than that of DP (average 68.5%). However, the recall of Hu (average 63.5%) is lower than both of DP (average 72.4%) and nDP (average 84.3%). In terms of F1 score, Hu and DP are likely

to be the same results (average 69.3% and 70.4% respectively) and less effective than nDP (average 80.4%). Our result analysis indicates that Hu's method is relatively effective in extracting frequent aspects with relatively high precision, but the disadvantage is that it just successfully extracts a small number of aspects which are frequent aspects (frequent items) in total number of correct aspects which includes infrequent aspects in the dataset. The higher recall figures of DP and nDP show that these methods extract infrequent aspects better than Hu's method. Overall, the precision, recall and F1 score of the proposed nDP method are mostly higher than those of two others, indicating the effectiveness of the nDP compared to DP and Hu algorithms.

Finally, the system generates the aspect-level review summary based on an input set of tuples  $(a, o, s)$  found in previous aspect extraction step and the proposed phase of aspect-level sentiment orientation and summary. For each discovered aspect of camera entity, each sentence in the input dataset which mentions this aspect is put into positive/negative/neural classes depending on the value of  $score(a_i, s_j)$  (Fig. 6).

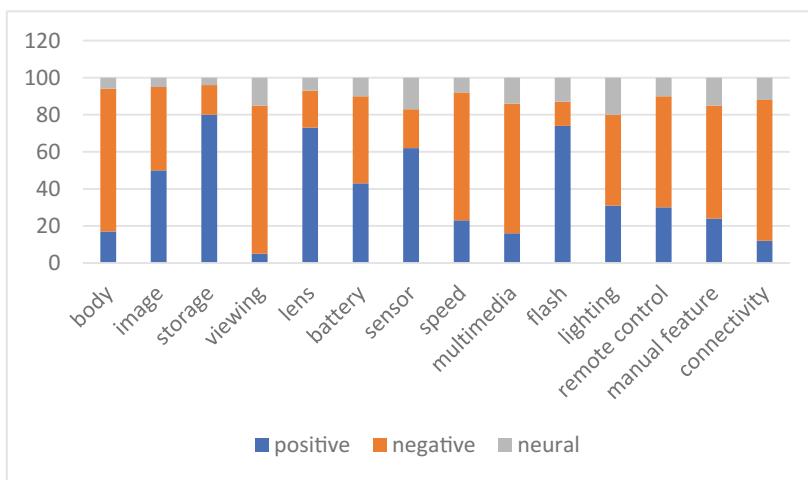


Fig. 6. Aspect-level opinion summary for experimental camera reviews

## 5 Conclusions

In this paper, we proposed some techniques for aspect extraction and sentiment analysis in aspect-based opinion mining problem, with the focus on: 1 - extracting both potential aspects and opinion words based on double propagation with some improvements to enhance the effectiveness of the model, 2 - classifying opinion about detected aspects in the context of review sentence using the hybrid approach of machine learning (Naïve Bayes classifier) and lexicon-based method (Wordnet) with the consideration of the sentence's context (dependency relations). Experimental results on the camera domain indicate that the proposed techniques are promising in performing the tasks of aspect-based opinion mining problem.

For future work, we plan to further improve and refine our techniques, and to address the challenging problems of determining the strength of opinions, and investigating opinions expressed with adverbs, verbs and nouns. We will also carry out more research and experiments in political and security domain.

## References

1. Liu, B.: *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, San Rafael ©2012
2. Khan, K., Baharudin, B., Khan, A.: Mining opinion components from unstructured reviews. *26*(3), 258–275 (2014)
3. Zhang, L., Liu, B.: Aspect and entity extraction for opinion mining. In: Chu, W. (ed.) *Data Mining and Knowledge Discovery for Big Data. SBD 2004*, vol. 1, pp. 1–40. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-642-40837-3\\_1](https://doi.org/10.1007/978-3-642-40837-3_1)
4. Maynard, D., Funk, A.: Automatic detection of political opinions in tweets. In: García-Castro, R., Fensel, D., Antoniou, G. (eds.) *ESWC 2011. LNCS*, vol. 7117, pp. 88–99. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-25953-1\\_8](https://doi.org/10.1007/978-3-642-25953-1_8)
5. Potha, N.: A biology-inspired, data mining framework for extracting patterns in sexual cyberbullying data. *Knowl.-Based Syst.* **96**, 134–155 (2016)
6. Zhao, R., Zhou, A., Mao, K.: Automatic detection of cyberbullying on social networks based on bullying features. In: *Proceedings of the 17th International Conference on Distributed Computing and Networking, ICDCN 2016*, Singapore (2016)
7. Sui, J.: Doctor of Philosophy: Understanding and fighting bullying with machine learning. University of Wisconsin-Madison (2015)
8. Lippmann, R.P., et al.: Toward finding malicious cyber discussions in social media. Presented at the The AAAI-17 Workshop on Artificial Intelligence for Cyber Security (2017)
9. Azizan, S.A., Aziz, I.A.: Terrorism detection based on sentiment analysis using machine learning. *J. Eng. Appl. Sci.* **12**, 691–698 (2017)
10. Wen, S., Haghghi, M.S., Chen, C., Xiang, Y., Zhou, W.L., Jia, W.J.: A sword with two edges: propagation studies on both positive and negative information in online social networks. *IEEE Trans. Comput.* **64**, 640–653 (2015)
11. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*
12. Hu, M., Liu, B.: Mining opinion features in customer reviews. Presented at the *AAAI 2004 Proceedings of the 19th National Conference on Artifical Intelligence*, pp. 755–760 (2004)
13. Popescu, A.-M., Etzioni, O.: Extracting product features and opinions from reviews. In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005*, pp. 339–346 (2005)
14. Qiu, G., Liu, B., Bu, J., Chen, C.: Expanding domain sentiment lexicon through double propagation. In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI 2009*, pp. 1199–1204 (2009)
15. Zhai, Z., Liu, B., Xu, H., Jia, P.: Grouping product features using semi-supervised learning with soft-constraints. In: *Proceedings of the 23rd International Conference on Computational Linguistics* (2010)
16. Raju, S., Shishtla, P., Varma, V.: Graph clustering approach to product attribute extraction. Presented at the *4th Indian International Conference on Artificial Intelligence* (2009)

17. Zhai, Z., Liu, B., Xu, H., Jia, P.: Clustering product features for opinion mining. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (2011)
18. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86 (2002)
19. Hatzivassiloglou, V., McKeown, K.: Predicting the semantic orientation of adjectives. In: Proceedings of Annual Meeting of the Association for Computational Linguistics (1997)
20. Wordnet. <https://wordnet.princeton.edu/>
21. Budanitsky, A., Hirst, G.: Semantic distance in wordnet: an experimental, application-oriented evaluation of five measures. Presented at the Workshop on WordNet and Other Lexical Resources (2001)
22. Hu, Liu: Opinion Lexicon: A list of English positive and negative opinion words or sentiment words. <https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon>
23. Ringsquandl, M., Petković, D.: Analyzing political sentiment on Twitter. In: 2013 AAAI Spring Symposium, Stanford University (2013)
24. Html Agility Pack. <http://html-agility-pack.net>
25. Stanford CoreNLP. <https://stanfordnlp.github.io/CoreNLP/>



# An Approach for Host-Based Intrusion Detection System Design Using Convolutional Neural Network

Nam Nhat Tran<sup>(✉)</sup>, Ruhul Sarker, and Jiankun Hu

University of New South Wales Canberra at the Australian Defence Force Academy,  
Canberra, Australia

[nam.tran@student.adfa.edu.au](mailto:nam.tran@student.adfa.edu.au), [{r.sarker,j.hu}@adfa.edu.au](mailto:{r.sarker,j.hu}@adfa.edu.au)

**Abstract.** Along with the drastic growth of telecommunication and networking, the cyber-threats are getting more and more sophisticated and certainly leading to severe consequences. With the fact that various segments of industrial systems are deployed with Information and Computer Technology, the damage of cyber-attacks is now expanding to physical infrastructure. In order to mitigate the damage as well as reduce the False Alarm Rate, an advanced yet well-design Intrusion Detection System (IDS) must be deployed. This paper focuses on system call traces as an object for designing a Host-based anomaly IDS. Sharing several similarities with research objects in Natural Language Processing and Image Recognition, a Host-based IDS design procedure based on Convolutional Neural Network (CNN) for system call traces is implemented. The decent preliminary results harvested from modern benchmarking datasets NGIDS-DS and ADFA-LD demonstrated this approachs feasibility.

**Keywords:** Intrusion Detection System · Host-Based  
Convolutional Neural Network

## 1 Introduction

The issues of cyber security have increasingly attracted the social concerns in recent decades. The catastrophic consequences of cyber-crimes are the main factor contributing to the development of security systems. Almost every regime of the contemporary digital society, such as industry, military, business, medicine, and so on, are involved with the cyber-infrastructure, it is now critical to protect the integrity of information. An Intrusion Detection System (IDS) is deployed with the purpose to provide a tool to monitor system and network operations against malicious activities and policy violations [20]. The IDS plays role as a guardian to classify the activities and then to trigger defense mechanisms, if necessary. The ultimate target of an IDS is to prevent cyber-attacks, or at least mitigate an ongoing attack. Thus, an IDS must theoretically possess sensible reaction, precise detection mechanism, and secure defense techniques against

malevolent attackers. From these points of view, an IDS is a crucial component that contributes to the procedure of forensic analysis in order to identify security breaches and vulnerabilities. Moreover, the potential of an IDS is not limited to detect cyber-attacks but it also expands to noticing abnormal system behavior to detect accidents or undesired conditions [39].

The quick development of the cyberspace has led to the booming in cyber-threats, which result in the increasing of various types of attacks. Therefore, an attack is either a defined one whose signature or pattern has already been discovered, or a brand new case with unknown signature. Based on the detection methods, IDSs are categorized into misuse detection and anomaly detection. Misuse or signature-based detection operates based on the principle of comparing the collected data with a database of known attack signatures in order to determine whether a pattern is matched. It is capable of detecting predefined threats but has no ability to cope with novel threats. More importantly, with the advance of technology, attackers are able to create polymorphic threats. This is a serious loophole of the signature-based detection techniques. Anomaly detection, however, is able to resolve the problem with unknown patterns by using machine learning algorithms to build a model for normal, trustworthy activities. Any excessive deviation from the normal model would be considered as a malicious threat, thus results in alerts for protection system. The main drawback of an anomaly IDS is its suffering from false positives such that a previously unknown legitimate operation would be classified as malicious. IDSs are also sorted out by where the detection takes place, forming the Network based IDS (NIDS) and the Host based IDS (HIDS). A network based IDS deploys one or several points of observation to collect and monitor inbound and outbound network traffics, then analyses such data to make decision. A host based IDS, which is considered as the final layer of defenses, works primarily on individual hosts or devices on a network. It monitors the critical components including but not limited to configurations, system logs, system processes, files, or network interfaces of the host, compare the current states with the previous stable states. Any modification that significantly deviates the system from the normal state brings up an alert for further action. Among those various types of systems raw data, system call traces have been extensively used as a specific measure for security evaluation since [13,14]. As a low-level direct interaction with the Unix-based systems kernel, system call traces could provide rich source of activity meanings, thus it is a top list priority object for security monitoring purpose [16,17].

It is the broad yet deep prevalence of the Internet and Computer Technology makes the traditional signature-based security resolution obsolete in dealing with such polymorphism. Therefore, anomaly-based IDS is an indisputable choice as the IDS step-by-step becomes an indispensable part of an ICT infrastructure. Various detection paradigms have been proposed, that were built based on classification techniques, statistical theories, information theories, and so on [3]. Some modern concepts also contribute to the increased complexity of the problem. The brilliant emerging of the next generation cellular networks and the introduction of a new class of client, Internet of Things (IoTs) are amongst

the evidences to demonstrate that an attack could be established from everywhere. In addition, the network of sensors also poses a new challenge on the traditional security and intelligence system with a new burden in terms of data, so called Big Data [21]. The enormous amount of data from fragmented sources is inherently a huge obstacle to the effort in searching for a comprehensive yet efficient security solution. In summary, the task for constructing comprehensive IDS framework solution is now required to incorporate many more techniques as well as to consider many other aspects of the problem. Machine learning classification techniques such as Neural Network is always considered first thanks to its potential in dealing with the constantly dynamic and evolving network threats. Especially, one of Neural Network branch, CNN is very effective when dealing with classification given enormous amount of data [26]. The decent results achieved recently with CNN through applications in NLP and Image Recognition has inspired this research. This paper is organized as follow: the first part presents introduction while the second part is dedicated to reviewing related works in the literatures; the third part discusses the research methodology, and the fourth one demonstrations works preliminary results; finally, conclusion and future research is drawn in the fifth part.

## 2 Related Work

The ability of malicious cyber threat to transform and protect itself from the signature-based IDS has led to the emerging of anomaly-based tools. Despite the fact that those machine learning based techniques could possibly introduce some false alarms, their provision of unknown detection is extremely useful to deal against polymorphic mechanism [4]. Several machine learning techniques are used extensively, includes but not limited to Support Vector Machine [18, 25, 30], Neural Network [4, 29], k-means Clustering (kMC) [37] and k-nearest neighbor (kNN) [27]. Among these, SVM which is based on the empirical risk minimization principle, has very high reputation as a popular and handy classification while Neural Network is gradually becoming a favor choice in IDS design thanks to its flexibility and effective classifying. The work of authors from [37] which is based on transforming system call traces into frequency vectors before reducing the dimension using Principle Component Analysis is conducted on the same ADFA-LD dataset as in this research. Therefore, these work that had conducted with kMC algorithm is correspondingly suitable to compare with our proposed work here, which fundamentally is also based on system call traces from ADFA-LD dataset.

A Convolutional Neural Network is fundamentally a Neural Network whose raw input data is divided into sub-regions and then using a specified number of filters to perform convolution before feeding through an activation function to build feature maps. The filters in convolutional layer has fixed size and will be slide horizontally then vertically (or vice versa) on “the surface” of input data until every sub-regions are covered. A non-linear activation function that is applied in the next step is typically a ReLU (*Rectified Linear Unit*)  $f(x) = \max(0, x)$  in

order to improve training performance on large and complex datasets [33]. On the next step, a pooling layer is deployed for down-sampling purpose (either max- or average-pooling) that could help to increase performance efficiency. The combination of three stacked layers convolutional, ReLU and pooling may be repeated in case of extracting features from high dimensional input data. After that, a fully connected layer, whose every node is connected to all the nodes from previous layer, is deployed to perform classification task. A softmax layer is also used to convert classification results into probabilities i.e., how likely an object is classified into one class.

The brilliant success of Convolutional Neural Network in classifying tasks from various technology giants like Google [23] or Facebook [2, 19] has inspired the idea of using CNN for designing a multiple classification IDS. Without mentioning the achievements from industrial firms, CNN has its own reputation for applications in image processing [6, 7, 11, 26] as well as natural language processing [8, 15]. However, it must be noted that the approaching method for image processing CNN application is differed from the way a CNN-based tool can handle an NLP task. This is due to the nature of input data for each application as the raw input data for an image processing application has at least two dimensions, thus convolutional filters (kernels) would be able to slide either vertically or horizontally to extract features from sub-regions of the input data; while the representation data for NLP applications have meaning along only one specified dimension, therefore the movement of the filters are strictly restricted. The case of input data for an IDS application is very similar to NLP applications, therefore, in order to exploit CNN for designing an IDS, input data sources must be “crafted and polished” into a suitable form. According to Ronan and Jason [8], a sentence can be transformed into a matrix by considering each word as a column vector that corresponding to an index from a finite dictionary. This rudimentary representation is then used as the input for a deep neural network. This method establishes a new way to approach CNN as any piece of information can possibly be digitalizing as a string of numbers. Hence, an observation or a group of observations from a training data set can be converted into a matrix during input design procedure. Sharing the same approach method, Zhang and Wallace in [38] also transformed a sentence of words into a matrix whose each row represents a word and the number of column is the dimension of set of word. One novel characteristic from this work is the employment of various-size filters in the convolutional layers before applying 1-max pooling layer to synchronize the result from previous layer. This strategy is a noticeable suggestion when it comes to apply for CNN-based HIDS design process.

### 3 Proposed Method

The main element to consider when analyzing host log files of a Unix-based system is the system calls. System call is defined as a request to the kernel that is generated by an ongoing process through interrupt mechanism [35]. This request is sent to the kernel because the active process needs to access some resources.

Therefore, a system call has a great impact on system state and is exactly an object under examining when it comes to system monitoring. System calls, however, are investigated as chain rather than a standalone one [5]. This is because a malicious activity normally contains a series of tasks, each of which has a specific request to the kernel. Thus, in order to identify an attack pattern, a series of several successive system calls and the related information are collected in a window and then analyzed. These windows are usually shifted chronologically, overlap each other with a predefined step for feature extraction purpose. This method was used extensively in several literatures, including [12, 22, 34].

Taking the main element under examining for a Host-based IDS as sequences of system calls is investigated through [12, 22, 34]. However, there is no general guideline or at least, a rule-of-thumb for the task of selecting adequate length for system call sequence. As this research is only at starting phase, taking on this problem heuristically is possibly a decent solution. Raw input data will be chunked into parts with length of a sliding window (which we are trying to find an optimized value heuristically) to form the input for training data as well as testing data for the CNN-based IDS model.

### 3.1 Data Sets

The datasets for benchmarking purpose here are Next Generation Intrusion Detection Systems Data Set (NGIDS-DS) and ADFA Linux Dataset (ADFA-LD), the two modern datasets were generated under the next generation cyber range infrastructure of the Australian Centre for Cyber Security at the Australian Defence Force Academy [31, 36]. Although there are several famous benchmarking data sets such as MIT Lincoln Laboratory's DARPA [10], KDD Cup 1999 Data [24], and NSL-KDD Dataset (an improved version of KDD'99) [32] etc., a few to names, ADFA-LD and NGIDS-DS were chosen for several reasons. First, these datasets were generated based on the modern computing infrastructures, contains up-to-date knowledge, thus it is able to reflect the latest characteristics and realistic performance of recent attacks. Second, the KDD'99 or DARPA are not only obsolete but also contains redundant information that might degrade the performance of the modern training system [28]. In addition, the NGIDS contains both host log files and .pcap (packet capture file), so it is convenient to analyze either HIDS or NIDS, or even perform a combined analysis. This is an important step toward building a comprehensive IDS, which is an indispensable part of the intelligence system for improving Critical Infrastructure Situational Awareness in future.

NGIDS-DS contains 99 host log files with csv format. Each record fully describes the relevant information about happened event, for both normal and malicious activities, including (Fig. 1) timestamp (date and time), event ID, path, process ID, system calls, attack category, attack subcategory, and label ("1" is marked for an attack and "0" is for a normal activity). A sliding window with fixed size will slide chronologically, extracts system calls and corresponding labels to form raw input data for CNN. The sliding windows may or may not overlap each other, which is also a parameter to determine. The principle of

making label for a sliding window is based on the fact that if a window contains any event marked “1”, the label for such window should be “1” also. Only when every activity inside a window are marked “0”, the label for that window is “0”. Figure 1 illustrates the sliding window method with size of 5 and no overlap. ADFA-LD is already divided into Attack, Training and Validation datasets. Each dataset contains various files of system call traces. Deploying the same sliding window approach as with NGIDS-DS, we also extracted raw input data for both training and testing phases successfully.

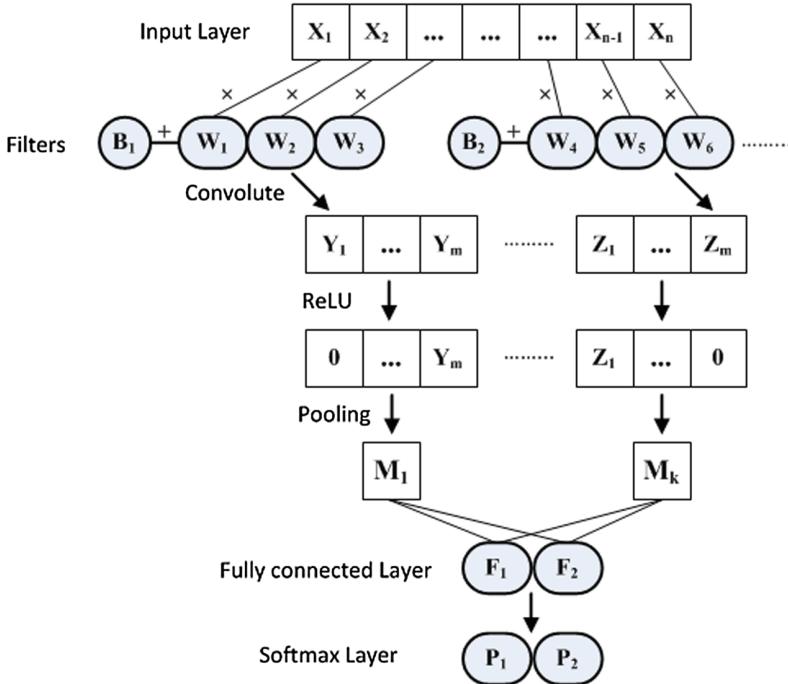
11/03/2016 2:45:01	1830	/sbin/upstart-dbus-bridge		142	45354	normal	normal	0
11/03/2016 2:45:01	1885	/usr/lib/unity/unity-panel-service	Window	168	45353	normal	normal	0
11/03/2016 2:45:01	1872	/usr/lib/unity/unity-panel-service		168	45355	normal	normal	0
11/03/2016 2:45:01	1951	/usr/lib/i386-linux-gnu/indicator-datetime/indicator-datetime-service		168	45350	normal	normal	0
11/03/2016 2:45:01	2114	/usr/bin/comptiz		168	45357	normal	normal	0
11/03/2016 2:45:01	1966	/usr/lib/i386-linux-gnu/indicator-datetime/indicator-datetime-service	Window	168	45351	normal	normal	0
11/03/2016 2:45:06	1804	/bin/dbus-daemon		256	45352	normal	normal	0
11/03/2016 2:45:06	2133	/usr/lib/i386-linux-gnu/gconfd-2		168	45372	normal	normal	0
11/03/2016 2:45:06	2834	/usr/bin/update-notifier		142	45360	normal	normal	0
11/03/2016 2:45:11	3989	/sbin/auditd		256	45374	normal	normal	0
11/03/2016 2:45:12	1086	/usr/lib/accountservice/accounts-daemon	Window	168	45362	normal	normal	0
11/03/2016 2:45:29	2106	/usr/lib/evolution/evolution-calendar-factory		168	45012	normal	normal	0
11/03/2016 2:45:29	2346	/usr/lib/telepathy/mission-control-5		168	45003	normal	normal	0
11/03/2016 2:45:29	1089	/usr/bin/whoopie		168	45007	normal	normal	0
11/03/2016 2:45:29	2264	/usr/lib/i386-linux-gnu/unity-scope-home/unity-scope-home		168	41361	normal	normal	0
11/03/2016 2:45:29	4009	/usr/lib/i386-linux-gnu/deja-dup/deja-dup-monitor		168	45008	normal	normal	0

**Fig. 1.** NGIDS-DS with sliding window for extracting data for CNN-based IDS training and testing phases

### 3.2 Convolutional Neural Network Based IDS

As mentioned above, CNN has been widely used for visual recognition and natural language processing, which are highly classification-oriented application. However, CNN is unprecedentedly applied to an IDS design procedure, even the input design is not straightforward as the others. Despite that challenge, CNN is undoubtedly suitable for this research in terms of availability of data. Since a CNN training phase requires a huge amount of data [26], the NGIDS-DS and ADFA-LD with up to hundreds of million observations are more than enough to work on. The issue with input data incompatibility was solved by using 4-dimensional (4-D) arrays, a popular feature that is available for frameworks like MATLAB [9] or TensorFlow [1]. An observation from raw input data will be transformed into an element of 4-D array, whose the first two dimensions are matrix sizes with the number of row is inherently one, and the number of column is equal to the size of sliding window. The third dimension of 4-D array is one, equivalently to be the channel in RGB image. The fourth dimension is the number of observation in the data set. This technique proved to work seamlessly with Matlab as well as TensorFlow.

For the sake of simplicity, time saving and due to the property of input data, the CNN architecture was merely deployed as follow with one convolutional layer only:



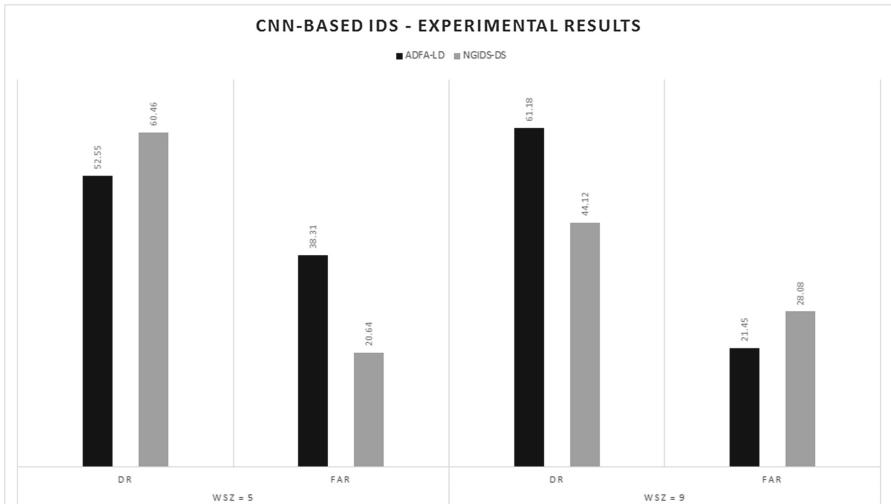
**Fig. 2.** Convolutional neural network architecture for Host-based IDS

The “Input Layer” has size of either [1 9 1] or [1 5 1], with 9 and 5 is the two best window sizes, which was determined heuristically. With such input data, the size of Filters (kernels) is also [1  $x$ ] with  $x$  is adjusted accordingly. Applying each filter respectively to the input entity, performs element-wise product between an input value  $\mathbf{X}$  with the filter’s weight  $\mathbf{W}$  then adds a bias  $\mathbf{B}$  to the result, an scalar value  $\mathbf{Y}$  is yielded. Sliding the filter along the input by a fix stride (step) produces a string of scalar value with the same height as the input data but smaller width. In the next step, those strings of number will be fed through a ReLU layer in order to introduce nonlinearities to the model. A ReLU keeps the same all non-negative values while to replace any input smaller than zero by zero. Then, a pooling layer (either a Max Pooling or Average Pooling depends on whichever provides a better performance) is applied for down-sampling data (a 1-pooling layer, which results in a scalar output, is demonstrated in Fig. 2). A fully connected layer whose each element is connected with all elements of the previous layer, is armed in order to perform classification based on features extracted from these previous layers. At this stage, a dropout layer with changeable drop rate is also deployed for solving over-fitted situations. Finally, softmax layer which plays role as a medium to convert the classification results into probabilities, is presented. The training process was conducted with *Stochastic Gradient Descent with Momentum* (SGDM) algorithm. With SGDM, the size of mini batch is also

an adjustable parameter, which was chosen based on optimized performance at the expense of training time. Two regularization methods L1 and L2 were also applied to enhance the experimental results.

## 4 Experimental Results and Analysis

The training and testing processes were conducted repeatedly through different sets of parameters. The two best sets of results are achieved with window size of 5 and 9. The Detection Rate (DR) which is calculated as the ratio of successfully detected abnormal events and the total number of abnormal events, and the False Alarm Rate (FAR) which is the average of the false positive rate and the false negative rate, are shown in Fig. 3. The temporary maximum DR is 61.18%, achieved through running ADFA-LD dataset with window size of 9. Meanwhile, the best DR value from NGIDS-DS is 60.46%, but with window size of 5. In conjunction with the detection rate of 60.46% for NGIDS-DS is the FAR 20.64%. Besides, the best DR from ADFA-LD is 61.18%. These results are somehow better than the counterpart of the work in [37] with DR of 60% and FAR of 20%. In addition, the kMC technique used in that literature is now considered obsolete due to two following reasons: First, in the Big Data era, the rapid growth in size of data significantly increases the computational cost for kMC. Secondly, as new data appears, the kMC-based algorithm needs remodelling, which means that training must be conducted again. The Neural Network approach, on the other hand, demonstrates its advantage in dealing with large-scale data, while maintaining the similar, competitive experimental results.



**Fig. 3.** The CNN-based IDS experimental results with different window sizes

## 5 Conclusion

The preliminary work in this paper has introduced a novel yet feasible approach method for Host-based Intrusion Detection System design. The design product worked well with large-scale raw input data, provided several decent experimental results from an extremely simple yet minimalistic architecture. This research has extended the field of application for Convolution Neural Network to a completely new regime, which, at first is seemed to be irrelevant. The detection rate as well as false alarm rate would possibly be improved by applying a more complicated model for CNN. As the neural network is inherently about feature extraction, a complex model would possibly produce more unique features, which in turn could certainly improve the classification results.

## References

1. A Guide to TF Layers: Building a Convolutional Neural Network. <https://www.tensorflow.org/tutorials/layers>. Accessed 08 Mar 2017
2. A path to unsupervised learning through adversarial networks. <https://code.facebook.com/posts/1587249151575490/a-path-to-unsupervised-learning-through-adversarial-networks/>. Accessed 03 Mar 2017
3. Ahmed, M., Mahmood, A.N., Hu, J.: A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **60**, 19–31 (2016)
4. Ashfaq, R.A.R., et al.: Fuzziness based semi-supervised learning approach for intrusion detection system. *Inf. Sci.* **378**, 484–497 (2017)
5. Canzanese, R., Mancoridis, S., Kam, M.: System call-based detection of malicious processes. In: 2015 IEEE International Conference on Software Quality, Reliability and Security (QRS), pp. 119–124. IEEE (2015)
6. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3642–3649. IEEE (2012)
7. Ciresan, D.C., et al.: Convolutional neural network committees for handwritten character classification. In: 2011 International Conference on Document Analysis and Recognition (ICDAR), pp. 1135–1139. IEEE (2011)
8. Collobert, R., Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning, pp. 160–167. ACM (2008)
9. Convolutional Neural Networks Matlab Documentation. <https://au.mathworks.com/help/nnet/convolutional-neural-networks.html>. Accessed 08 Mar 2017
10. DARPA Intrusion Detection Data Sets. <https://www.ll.mit.edu/ideval/data/>. Accessed 28 Feb 2017
11. Egmont-Petersen, M., de Ridder, D., Handels, H.: Image processing with neural networks—a review. *Pattern Recogn.* **35**(10), 2279–2301 (2002)
12. Fan, S., et al.: A dynamic on-line sliding window support vector machine for tunnel settlement prediction. In: 2013 3rd International Conference on Computer Science and Network Technology (ICCSNT), pp. 547–551. IEEE (2013)
13. Forrest, S., Hofmeyr, S., Somayaji, A.: The evolution of system-call monitoring. In: Annual Computer Security Applications Conference, ACSAC 2008, pp. 418–430. IEEE (2008)

14. Forrest, S., et al.: A sense of self for unix processes. In: Proceedings of 1996 IEEE Symposium on Security and Privacy, pp. 120–128. IEEE (1996)
15. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649. IEEE (2013)
16. Hoang, X.D., Hu, J., Bertok, P.: A multi-layer model for anomaly intrusion detection using program sequences of system calls. In: Proceedings of 11th IEEE International Conference. Citeseer (2003)
17. Hofmeyr, S.A., Forrest, S., Somayaji, A.: Intrusion detection using sequences of system calls. *J. Comput. Secur.* **6**(3), 151–180 (1998)
18. Horng, S.-J., et al.: A novel intrusion detection system based on hierarchical clustering and support vector machines. *Expert Syst. Appl.* **38**(1), 306–313 (2011)
19. Introducing DeepText: Facebook’s text understanding engine. <https://code.facebook.com/posts/181565595577955/introducing-deep-text-facebook-s-text-understanding-engine/>. Accessed 03 Mar 2017
20. Intrusion Detection System. [https://en.wikipedia.org/w/index.php?title=Intrusion\\_detection\\_system](https://en.wikipedia.org/w/index.php?title=Intrusion_detection_system). Accessed 30 Nov 2016
21. Jaradat, M., et al.: The internet of energy: smart sensor networks and big data management for smart grid. *Procedia Comput. Sci.* **56**, 592–597 (2015)
22. Kaneda, Y., Mineno, H.: Sliding window-based support vector regression for predicting micrometeorological data. *Expert Syst. Appl.* **59**, 217–225 (2016)
23. Karpathy, A., et al.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732 (2014)
24. KDD Cup 1999 Data. <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>. Accessed 28 Feb 2017
25. Khan, L., Awad, M., Thuraisingham, B.: A new intrusion detection system using support vector machines and hierarchical clustering. *VLDB J. Int. J. Very Large Data Bases* **16**(4), 507–521 (2007)
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
27. Liao, Y., Vemuri, V.R.: Use of k-nearest neighbor classifier for intrusion detection. *Comput. Secur.* **21**(5), 439–448 (2002)
28. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6. IEEE (2015)
29. Mukkamala, S., Janoski, G., Sung, A.: Intrusion detection using neural networks and support vector machines. In: Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN 2002, vol. 2, pp. 1702–1707. IEEE (2002)
30. Mukkamala, S., Sung, A.H.: Detecting denial of service attacks using support vector machines. In: The 12th IEEE International Conference on Fuzzy Systems, FUZZ 2003, vol. 2, pp. 1231–1236. IEEE (2003)
31. Next Generation Intrusion Detection Systems Data Set (NGIDS-DS): Overview. [https://research.unsw.edu.au/sites/all/files/facultyadmin/ngids-ds\\_overview\\_final.pdf](https://research.unsw.edu.au/sites/all/files/facultyadmin/ngids-ds_overview_final.pdf). Accessed 28 Feb 2017
32. NSL-KDD Data Set. <http://www.unb.ca/cic/research/datasets/nsl.html>. Accessed 28 Feb 2017
33. Rectifier (neural networks). [https://en.wikipedia.org/wiki/Rectifier\\_\(neural\\_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)). Accessed Mar 2017

34. Suzuki, Y., et al.: Proposal to sliding window-based support vector regression. *Procedia Comput. Sci.* **35**, 1615–1624 (2014)
35. System Call Definition. [http://www.linfo.org/system\\_call.html](http://www.linfo.org/system_call.html). Accessed 01 Feb 2017
36. The ADFA Linux Dataset (ADFA-LD). <https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-IDS-Datasets/>. Accessed 28 Feb 2017
37. Xie, M., Hu, J., Yu, X., Chang, E.: Evaluating host-based anomaly detection systems: application of the frequency-based algorithms to ADFA-LD. In: Au, M.H., Carminati, B., Kuo, C.-C.J. (eds.) NSS 2014. LNCS, vol. 8792, pp. 542–549. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-11698-3\\_44](https://doi.org/10.1007/978-3-319-11698-3_44)
38. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In: arXiv preprint [arXiv:1510.03820](https://arxiv.org/abs/1510.03820) (2015)
39. Zuech, R., Khoshgoftaar, T.M., Wald, R.: Intrusion detection and big heterogeneous data: a survey. *J. Big Data* **2**(1), 3 (2015)



# A Robust Contactless Fingerprint Enhancement Algorithm

Xuefei Yin, Yanming Zhu, and Jiankun Hu<sup>(✉)</sup>

School of Engineering and Information Technology,  
University of New South Wales, Canberra, ACT 2600, Australia  
 [{xuefei.yin,yanming.zhu}@student.unsw.edu.au](mailto:{xuefei.yin,yanming.zhu}@student.unsw.edu.au), [J.Hu@adfa.edu.au](mailto:J.Hu@adfa.edu.au)

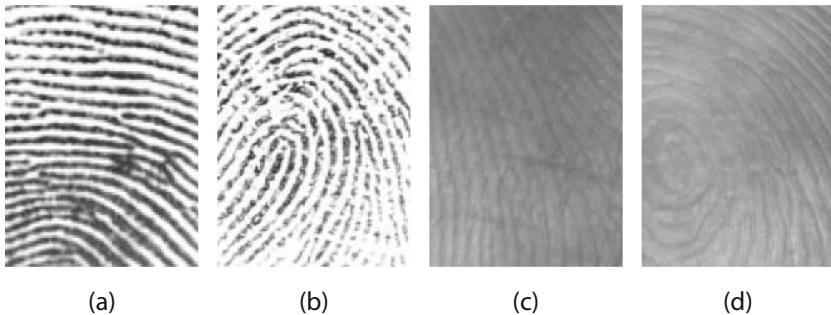
**Abstract.** Compared to contact fingerprint images, contactless fingerprint images have three particular characteristics: (1) contactless fingerprint images have less noise than contact fingerprint images; (2) there are less discontinuities of ridges in contactless fingerprint images; and (3) the ridge-valley pattern of contactless fingerprint is much more unclear than that of contact fingerprint images. These properties increase a great difficulty to the contactless fingerprint enhancement. In this paper, we propose a robust contactless fingerprint enhancement algorithm based on simple sinusoidal-shaped filter kernel to fully take advantage of the properties of contactless fingerprint. First, an effective preprocessing is proposed to preliminarily strengthen the ridge-valley contrast of contactless fingerprint images. Then, simple sinusoidal-shaped filter kernel is proposed to enhance the contactless fingerprint images. Finally, we propose a score-filtering procedure to effectively recover the ridge-valley pattern. Comprehensive experiments were performed to evaluate the proposed method from aspects of image quality, minutiae extraction and fingerprint verification. Experimental results demonstrate the high performance of the proposed algorithm in contactless fingerprint enhancement.

**Keywords:** Contactless fingerprint enhancement  
Sinusoidal-shaped filter · Ridge orientation · Ridge frequency  
Fingerprint

## 1 Introduction

With the development and popularity of sensing technologies, contactless biometrics technologies (e.g., identification and verification) have become a hot research area and have attracted great attentions in commercial applications [8, 10, 11, 13, 22]. The National Institute of Standards and Technology (NIST) has announced the plan to develop *Next Generation Fingerprint Technologies*. In this project, the contactless fingerprint technology is one of the most important parts, which demonstrates the highly promising prospects of contactless fingerprint technologies in the future. Compared with the contact fingerprint

collection, contactless fingerprint caption system can avoid many risks such as image contamination, time-consuming issue, nonlinear distortion and hygienic concern. However, contactless fingerprint images tend to suffer from low ridge-valley contrast. Figure 1 gives an example of the two types of fingerprint images. As shown in the figure, the ridge-valley contrast of the contactless fingerprint is quite unclear compared to that of contact fingerprint. This would badly affect the performance of minutiae extraction progress [5] and other subsequent progresses [16, 20]. Therefore, it is essential to develop a reliable and effective technique for contactless fingerprint enhancement.



**Fig. 1.** Examples of two types of fingerprint images: (a) and (b) are partial images of contact fingerprints from FVC2004 DB1 [9], (c) and (d) are partial images of contactless fingerprints from public dataset [22].

Over the past few decades, many methods have been proposed to enhance the ridge-valley contrast of fingerprint images and have made a remarkable progress. According to the filtering domain, the enhancement methods can be basically categorized into two classes: (1) spatial domain filtering [7, 12, 19, 21, 23] and (2) Fourier domain filtering [4, 15, 18].

Among the spatial domain filtering methods, contextual filters are the most widely used for fingerprint image enhancement. Nickerson and O’Gorman [12] firstly introduced contextual filters for fingerprint enhancement. The shape of those filters is controlled by the ridge frequency and the ridge orientation. However, in order to reduce computational complexity, the local ridge frequency is assumed constant in this method, leading to imprecise filtering result in some regions. Hong et al. [7] proposed an effective enhancement method based on Gabor filter, whose shape is controlled by four parameters. The advantage of this method is that the frequency and the orientation of the filter are adaptively determined by the local ridge frequency and the local ridge orientation. However, the filtering result is poor in regions where the fingerprint ridge and valley pattern are not similar with a pure sinusoidal pattern. In order to address this problem, Yang et al. [19] proposed to use positive and negative ridge frequencies based on the local ridge width and local valley width, respectively. Compared to squared Gabor filter kernel in [7, 19], Zhu et al. [23] proposed using circular filter

kernel, which is helpful to avoid artifacts in the filtering progress. The above-mentioned methods are mainly focused on contact fingerprint enhancement. A novel method which focuses on contactless fingerprint enhancement is proposed by Yin et al. [21]. In this method, intrinsic image decomposition [14] and guided image filtering [6] are firstly introduced for contactless fingerprint enhancement. However, this method tends to fail in the regions near singularity points.

Besides spatial domain filtering methods, Fourier domain filtering is another widely used technique for fingerprint enhancement. In these methods, filters are explicitly defined in the Fourier domain. Sherlock et al. [15] proposed using Fast Fourier Transform to enhance fingerprint images. In this method, the Fourier transform of the fingerprint image is multiplied by  $n$  precomputed filters. The pixel value of enhancement fingerprint is determined by the result of the filter whose orientation is closest to the local ridge orientation. The drawback of this method is that the ridge frequency is constant. Watson et al. [18] proposed an enhancement method in the Fourier domain, where the local ridge frequency and the local ridge orientation are no need to compute explicitly. However, this method is time-consuming because a large amount of overlap is introduced between the neighboring blocks. Chikkerur et al. [4] proposed an efficient enhancement method based on short-time Fourier transform (STFT). The enhancement result of the method is relatively similar to that of the method in [7]. The advantage of this method is that it costs less time than the method in [7] while the disadvantage is that this method tends to fail in the regions near singularity points.

Most of above-mentioned methods are proposed to enhance contact fingerprint images. These methods do not take the properties of contactless fingerprint images into account. Compared to contact fingerprint images, contactless fingerprint images have three particular properties: (1) contactless fingerprint images have less noise than contact fingerprint images, as shown in Fig. 1; (2) there are less discontinuities of ridges in contactless fingerprint images, which is helpful for the filtering process to enhance fingerprint images; and (3) the ridge-valley contrast of contactless fingerprint is much more unclear than that of contact fingerprint images, which increases a great difficulty to the contactless fingerprint enhancement. Based on the above analysis of contact fingerprint enhancement methods and in order to fully take advantage of the contactless fingerprint properties, in this paper, we propose a robust contactless fingerprint enhancement algorithm based on simple sinusoidal-shaped filter kernel. The main contributions of this paper are summarized as follows. First, an effective preprocessing is proposed to strengthen the ridge-valley contrast of contactless fingerprint images. Second, simple sinusoidal-shaped filter kernel is proposed to enhance contactless fingerprint images. Third, we propose score-filtering procedure to effectively recover the ridge-valley pattern. Experimental results demonstrate the validity of the proposed method in contactless fingerprint enhancement.

The rest of this paper is structured as follows: Sect. 2 describes the proposed approach. Experimental validation and results are presented in Sect. 3. Finally, Sect. 4 concludes the paper.

## 2 The Proposed Method

In this section, we present the details of the proposed contactless fingerprint enhancement algorithm, which contains the following main steps: contactless fingerprint image preprocessing (Sect. 2.1), dominant ridge orientation estimation (Sect. 2.2), local ridge frequency estimation (Sect. 2.3), filtering (Sect. 2.4) and score-filtering procedure (Sect. 2.5).

### 2.1 Image Preprocessing

Compared with contact fingerprint images whose ridge-valley pattern is relatively clear, contactless fingerprint images tend to have a low ridge-valley contrast in small local regions. This negatively affects the filtering result. Image preprocessing is aimed at stretching ridge-valley contrast initially and hence facilitates effective filtering in the subsequent process.

As the pixel intensity varies considerably in different regions of contactless fingerprint images, it is unsuitable to perform global image enhancement technique on entire image. In this paper, we propose a region-based technique to preprocess contactless fingerprint images. First, contrast-limited adaptive histogram equalization [24], which is an effective region-based method to improve regions' contrast, is used to stretch the ridge-valley contrast of contactless fingerprint images. Then, each small region is normalized according to Eq. (1)

$$p'_i = (p_i - \mu)/S, \quad (1)$$

where  $\mu$  and  $S$  are the pixel mean value and standard deviation in a small region, respectively.  $p_i$  and  $p'_i$  are the pixel value and transformed pixel value, respectively. In order to eliminate artificial boundaries, the normalized neighboring regions are then combined using bilinear interpolation.

### 2.2 Dominant Ridge Orientation Estimation

Since contactless fingerprint image has better ridge continuity quality than contact fingerprint image, in this paper, we use the gradient-based method [2, 7, 17] to estimate the dominant ridge orientation. First, we calculate the x-gradient ( $G_x$ ) and the y-gradient ( $G_y$ ) based on the preprocessed image. In order to avoid orientation average problem, doubling the angle of gradient vector and squaring the length of gradient vector are performed as Eq. (2)

$$\begin{bmatrix} G_{sx} \\ G_{sy} \end{bmatrix} = \begin{bmatrix} \sqrt{G_x^2 + G_y^2}^2 \cos 2\theta \\ \sqrt{G_x^2 + G_y^2}^2 \sin 2\theta \end{bmatrix} = \begin{bmatrix} G_x^2 - G_y^2 \\ 2G_x G_y \end{bmatrix}, \quad (2)$$

where  $[G_{sx}, G_{sy}]^T$  denotes the squared gradient vector. Since the dominant ridge orientation varies steadily in a small local regions,  $G_{sx}$  and  $G_{sy}$  is therefore smoothed by a Gaussian filter of radius  $r_0$  pixels with standard deviation  $\sigma_0$ .

Then, the average squared gradient  $[\bar{G}_{sx}, \bar{G}_{sy}]^T$  in a local window  $W$  is calculated as Eq. (3)

$$\begin{bmatrix} \bar{G}_{sx} \\ \bar{G}_{sy} \end{bmatrix} = \begin{bmatrix} \sum_W G_{sx} \\ \sum_W G_{sy} \end{bmatrix} = \begin{bmatrix} \sum_W (G_x^2 - G_y^2) \\ \sum_W 2G_x G_y \end{bmatrix}. \quad (3)$$

Finally, the dominant ridge orientation  $\theta$  at pixel  $i$  is given by Eq. (4)

$$\theta_i = \frac{\pi}{2} + \frac{\text{atan}2(\bar{G}_{sy}, \bar{G}_{sx})}{2}. \quad (4)$$

### 2.3 Local Ridge Frequency Estimation

Since contactless fingerprint images have less noise than contact fingerprint images, it is easily to compute the average number of pixels between two consecutive peaks in a sinusoidal-shaped ridge-valley pattern. In this paper,  $x$ -signature method [7] is used to estimate the local ridge frequency. Let  $W$  denotes the rectangle region oriented by  $\theta_i$  degree, where  $\theta_i$  is the dominant ridge orientation at the center of the region, the average length  $l_i$  can be computed as Eq. 5

$$l_i = \sum (p_{j+1} - p_j)/(N - 1), \quad (5)$$

where  $p_j$  is the position of the  $j$ <sup>th</sup> peak,  $N$  is the number of consecutive peaks in the rectangle region. The local ridge frequency  $f_i$  in a local region is presented by

$$f_i = 1/l_i. \quad (6)$$

### 2.4 Filtering

Compared to contact fingerprint images, contactless fingerprint images have two different properties: (1) good sinusoidal-shaped pattern quality, which is less contaminated by image noise and (2) better ridge continuity quality than contact fingerprint images. Therefore, in order to take fully advantage of the two properties, in this paper, we propose using a simple sinusoidal-shaped filter kernel to effectively improve the ridge-valley contrast. The sinusoidal-shaped filter kernel is formulated as

$$k(x, y; \theta, f) = \cos(2\pi f \cdot (x \cos \theta + y \sin \theta)), \quad (7)$$

where  $\theta$  and  $f$  are the orientation and the frequency, respectively.  $x, y$  are the coordinates. Given a filter of size  $(2r_1 + 1) \times (2r_2 + 1)$ , the filtering process can be expressed as

$$I'(x, y) = \sum_{s=-r_1}^{r_1} \sum_{t=-r_2}^{r_2} k(s, t; \theta_{xy}, f_{xy}) I(x + s, y + t), \quad (8)$$

where  $\theta_{xy}$  is the dominant ridge orientation and  $f_{xy}$  is the local ridge frequency.

## 2.5 Score-Filtering Procedure

In order to effectively improve the ridge-valley contrast of contactless fingerprint images, we propose a score-filtering procedure which has two advantages: (1) eliminating artificial boundaries accused by block filtering and (2) avoiding disconnectedness in the regions near singularity points and minutiae.

First, the preprocessed image is divided into overlap blocks with size  $(2r + 1) \times (2r + 1)$  with  $r$  pixels overlapping in  $x$  or  $y$  direction. Then, for each small block  $\mathbf{b}_i$  centered at pixel  $(x_i, y_i)$ , the filtering result of the central pixel is calculated as Sect. (2.4) while the filtering result of each pixel is determined by the filtering result of the central pixel, which is formulated as

$$I'(\mathbf{b}_i) = I'(x_i, y_i). \quad (9)$$

Since each pixel is covered by  $N$  blocks, the final-filtering result of each pixel is calculated by

$$\bar{I}(x, y) = \frac{\sum I'(x_i, y_i)}{N}, \quad (10)$$

where  $(x_i, y_i)$  is the center of each covering block.

## 3 Experimental Results

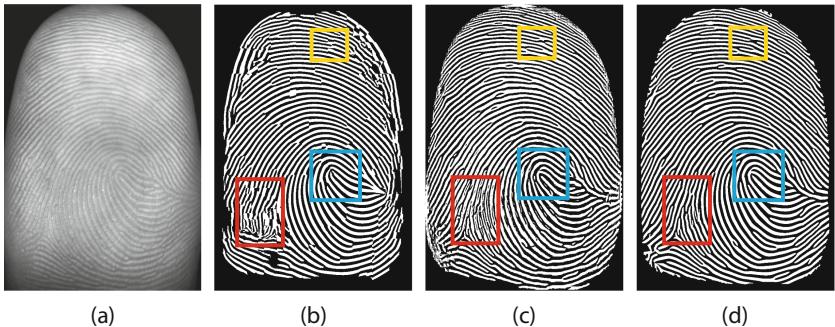
In this section, we evaluate the performance of the proposed algorithm in terms of fingerprint image quality (in Sect. 3.1), minutiae extraction (in Sect. 3.2) and fingerprint verification (in Sect. 3.3). The experiment is evaluated on contactless fingerprint benchmark database [22], which contains 1500 objects. In the experiment, we compare the proposed algorithm with the best two enhancement methods, traditional Gabor-based method (TGM) [7] and short-time Fourier transform method (STFT) [4]. Parameter values used in the proposed algorithm are reported in Table 1 while parameter values for the other two algorithms are directly employed from the papers.

**Table 1.** Parameter values used in the proposed algorithm

Symbol(s)	Value(s)	Description
$r_0$	25	Parameter in Sect. 2.2
$\sigma_0$	8	Parameter in Sect. 2.2
$r_1, r_2$	$1/f_{xy}$	Parameters in Sect. 2.4
$r$	5	Parameter in Sect. 2.5

### 3.1 Evaluation on Fingerprint Image Quality

In this section, we evaluate the performance of the proposed algorithm in terms of fingerprint image quality. This experiment is evaluated on contactless fingerprint images from publicly fingerprint database [22]. We compare the performance of the proposed algorithm with the best two enhancement methods, TGM [7] and STFT [4]. Figure 2 shows the comparison of enhancement results evaluated on contactless fingerprint image sample F3620. As shown in the figure, the proposed algorithm precisely strengthens the ridge-valley contrast in the entire image, while TGM and STFT fail in some regions, for example in the red rectangle and yellow rectangle regions in Fig. 2(b) and (c). Moreover, the proposed algorithm achieves better enhancement performance in the region near singularity point than the other two methods, as shown in the blue rectangle regions in the figure. In the blue rectangle region, the proposed algorithm accurately keeps ridge continuity while the other two methods result in bad ridge discontinuity.



**Fig. 2.** Comparison of contactless fingerprint image enhancement results on sample F3620: (a) Original contactless fingerprint, (b) TGM method [7], (c) STFT method [4], and (d) The proposed algorithm. (Color figure online)

### 3.2 Evaluation on Minutiae Extraction

In this section, we evaluate the performance of the proposed algorithm in terms of minutiae extraction. This experiment is evaluated on 100 contactless fingerprint images randomly selected from the database [22]. The commercial fingerprint software Verifinger SDK [1] is used for the minutiae extraction. In order to accurately compare minutiae extraction results, the minutiae are manually labelled in advance as the ground-truth. In this experiment, three measures are used to evaluate the accuracy of minutiae extraction:

- AGM: the average number of genuine minutiae, which are extracted on the enhanced fingerprint images but are not extracted on the original images.

**Table 2.** The comparison of average numbers of minutiae

	AGM	AFGM	AFM
TGM method [7]	5.1	2.8	18.3
STFT method [4]	5.8	1.9	10.1
Proposed method	6.7	0.5	3.6

- AMGM: the average number of genuine minutiae, which are not extracted on the enhanced fingerprint images but are extracted on the original images.
- AFM: the average number of detected minutiae, which are not genuine minutiae.

Table 2 shows the average number of three types of minutiae. As shown in Table 2, the proposed algorithm achieves better performance on the three measures than the other two methods. Compared to the other two methods, the proposed enhancement algorithm recover more genuine minutiae (AGM 6.7) and generates less false minutiae (AFM 3.6). This demonstrates that the proposed algorithm achieves remarkable performance.

### 3.3 Evaluation on Fingerprint Verification

In this section, we evaluate the performance of the proposed contactless fingerprint enhancement algorithm in terms of verification accuracy. MCC method [3] is used to perform the fingerprint verification. Three measures are used to evaluate the verification accuracy:

- False Matching Rate (FMR): the rate of different fingerprints which are decided to come from the same finger by a matching method.
- False Non-Matching Rate (FNMR): rate of corresponding fingerprints which are decided to come from the different fingers by a matching method.
- FMR100: the lowest FNMR at the threshold where  $FMR \leq 1\%$ .
- FMR1000: the lowest FNMR at the threshold where  $FMR \leq 0.1\%$ .
- Equal-Error Rate (EER): the error rate at the threshold where FMR and FNMR are equal.

**Table 3.** The comparison of verification accuracy

	EER %	FMR100 %	FMR1000 %
Without enhancement	9.73	12.58	14.08
With enhancement	5.8	7.22	9.32

Table 3 shows the experimental results of the three measures without enhancement and with enhancement using the proposed algorithm. As shown in the table, by using the proposed algorithm, the fingerprint verification achieves significant improvement in terms of EER, FMR100 and FMR1000.

## 4 Conclusion

In conclusion, this paper developed a robust contactless fingerprint enhancement algorithm, which takes full advantage of the special properties (less noise, low contrast and a good quality of ridge continuity). First, an effective preprocessing is proposed to preliminarily strengthen the ridge-valley contrast of contactless fingerprint images. Then, a simple sinusoidal-shaped filter kernel is proposed to enhance contactless fingerprint images. Finally, we proposed score-filtering procedure to effectively recover the ridge-valley pattern by eliminating artificial boundaries accused by block filtering and avoiding disconnectedness in the regions near singularity points and minutiae. The performance of the proposed algorithm is evaluated in terms of image quality, minutiae extraction and fingerprint verification. Experimental results show that the proposed algorithm considerably improves the performance of minutiae extraction and the performance of fingerprint verification.

**Acknowledgments.** The authors would like to thank the support from ARC project LP120100595.

## References

1. Verifinger SDK. <http://www.neurotechnology.com/verifinger.html>
2. Bazen, A.M., Gerez, S.H.: Systematic methods for the computation of the directional fields and singular points of fingerprints. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 905–919 (2002)
3. Cappelli, R., Ferrara, M., Maltoni, D.: Minutia cylinder-code: a new representation and matching technique for fingerprint recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(12), 2128–2141 (2010)
4. Chikkerur, S., Cartwright, A.N., Govindaraju, V.: Fingerprint enhancement using STFT analysis. *Pattern Recogn.* **40**(1), 198–211 (2007)
5. Farina, A., Kovacs-Vajna, Z.M., Leone, A.: Fingerprint minutiae extraction from skeletonized binary images. *Pattern Recogn.* **32**(5), 877–889 (1999)
6. He, K., Sun, J., Tang, X.: Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(6), 1397–1409 (2013)
7. Hong, L., Wan, Y., Jain, A.: Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(8), 777–789 (1998)
8. Kim, D., Jung, Y., Toh, K.A., Son, B., Kim, J.: An empirical study on iris recognition in a mobile phone. *Expert Syst. Appl.* **54**, 328–339 (2016)
9. Maio, D., Maltoni, D., Cappelli, R., Wayman, J.L., Jain, A.K.: FVC2004: third fingerprint verification competition. In: Zhang, D., Jain, A.K. (eds.) ICBA 2004. LNCS, vol. 3072, pp. 1–7. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-25948-0\\_1](https://doi.org/10.1007/978-3-540-25948-0_1)
10. Michael, G.K.O., Connie, T., Teoh, A.B.J.: A contactless biometric system using multiple hand features. *J. Vis. Communun. Image Represent.* **23**(7), 1068–1084 (2012)
11. Michael, G.K.O., Connie, T., Teoh Beng Jin, A.: An innovative contactless palm print and knuckle print recognition system. *Pattern Recogn. Lett.* **31**(12), 1708–1719 (2010)

12. Nickerson, J.V., O'Gorman, L.: An approach to fingerprint filter design. *Pattern Recogn.* **22**(1), 29–38 (1989)
13. Oh, B.S., Oh, K., Teoh, A.B.J., Lin, Z., Toh, K.A.: A gabor-based network for heterogeneous face recognition. *Neurocomputing* **261**, 253–265 (2017)
14. Shen, J.B., Yang, X.S., Li, X.L., Jia, Y.D.: Intrinsic image decomposition using optimization and user scribbles. *IEEE Trans. Cybern.* **43**(2), 425–436 (2013)
15. Sherlock, B.G., Monro, D.M., Millard, K.: Fingerprint enhancement by directional Fourier filtering. *IEE Proc. - Vis. Image Sig. Process.* **141**(2), 87–94 (1994)
16. Wang, Y., Hu, J., Phillips, D.: A fingerprint orientation model based on 2D Fourier expansion (FOMFE) and its application to singular-point detection and fingerprint indexing. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(4), 573–585 (2007)
17. Wang, Y., Hu, J., Han, F.: Enhanced gradient-based algorithm for the estimation of fingerprint orientation fields. *Appl. Math. Comput.* **185**(2), 823–833 (2007)
18. Watson, C.I., Candela, G.T., Grother, P.J.: Comparison of FFT fingerprint filtering methods for neural network classification. *NISTIR* **5493**, 1994 (1994)
19. Yang, J., Liu, L., Jiang, T., Fan, Y.: A modified gabor filter design method for fingerprint image enhancement. *Pattern Recogn. Lett.* **24**(12), 1805–1817 (2003)
20. Yang, W., Hu, J., Stojmenovic, M.: NDTC: a novel topology-based fingerprint matching algorithm using n-layer delaunay triangulation net check. In: Proceedings of the 2012 7th IEEE Conference on Industrial Electronics and Applications, ICIEA 2012, pp. 866–870 (2012)
21. Yin, X., Hu, J., Xu, J.: Contactless fingerprint enhancement via intrinsic image decomposition and guided image filtering. In: 2016 IEEE 11th Conference on Industrial Electronics and Applications, pp. 144–149 (2016)
22. Zhou, W., Hu, J., Petersen, I., Wang, S., Bennamoun, M.: A benchmark 3D fingerprint database. In: 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 935–940 (2014)
23. Zhu, E., Yin, J., Zhang, G.: Fingerprint enhancement using circular gabor filter. In: Campilho, A., Kamel, M. (eds.) ICIAR 2004. LNCS, vol. 3212, pp. 750–758. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-30126-4\\_91](https://doi.org/10.1007/978-3-540-30126-4_91)
24. Zuiderveld, K.: Contrast Limited Adaptive Histogram Equalization, pp. 474–485. Academic Press Professional, Inc., San Diego (1994)



# Designing Anomaly Detection System for Cloud Servers by Frequency Domain Features of System Call Identifiers and Machine Learning

Waqas Haider, Jiankun Hu, and Nour Moustafa<sup>(✉)</sup>

School of Engineering and Information Technology,  
UNSW Canberra, Canberra, Australia  
[nour.moustafa@unsw.edu.au](mailto:nour.moustafa@unsw.edu.au)

**Abstract.** The protection of operating systems from the current cyber threats has paramount importance. This importance is reflected by the functional dependency of any known or unknown cyber-attack upon the machines operating system. In order to design an anomaly detection system to protect an operating system from unknown attacks, acquiring comprehensive information related to running activities is the first crucial step. System call identifiers are one of the most reflective logs related to running activities in an operating system. Number of system call identifiers based host anomaly detection systems have been presented from the last two decades by using logs as raw system call identifiers. However, due to the stealth and penetration power of the unknown attacks, there is a need of acquiring and investigating more possible logs from machines operating system for the reliable protection. In this paper, firstly we apply the sine and Fourier transformation to the short sequence of system call identifiers, in order to model the frequency domain feature vector of any running activity at the cloud server. Second, different machine learning algorithms are trained and tested as anomaly detection engine using frequency domain transformed feature vectors of the short sequence of system call identifiers. The proposed work is evaluated using recently released intrusion detection systems data-set i.e., NGIDS-DS alongside two other old data-sets for comparative purposes. The experimental results indicate that the frequency domain feature vectors of short sequence of system call identifiers have comparatively superior performance than raw short sequence of system call identifiers, in detecting anomalies and building normal profile.

**Keywords:** HIDS · HADS · Operating system security  
Intrusion detection

## 1 Introduction

Although firewall technology [1] and access control mechanisms [2,3] can provide strong cybersecurity protection, the wide spread of advanced hacking tools

plus the daunting number of combinations of vulnerable points from software, operating systems and networking protocols has rendered it impossible to prevent all cyberattacks, in particular zero-day attacks [4, 5, 5, 6]. Today hacking groups which may be sponsored by the governments or individuals can design and launch the type of cyber-attacks which are capable of penetration through network defense zone [7–12]. Such type of attacks are only visible at machines operating system while performing the malicious tasks. The global cyber threats reports alarming the fact that, the target of these attacks are critical machines. For example, storage and processing servers in the cloud computing environment are prime targets, because at present corporate enterprises utilize cloud computing infrastructure for data to analyze, interpret and to make proactive decisions to keep the business competitive [13]. Further, most of the storage and processing servers in cloud computing infrastructure are comprised of Linux and Unix based operating systems [14]. During operation, the patterns of any legitimate or anomalous events in these operating systems are present at the kernel level system call identifiers sequences. Each system call identifiers sequence represents the relation of activity resource consumption at the software level with the time [15].

Detecting anomalous behavior in critical cloud servers has been observed to be a serious problem for the cloud computing service providers, due to the following two major reasons: (i) During the last two decades, number of system call identifiers based host intrusion detection systems are presented [16, 17]. In these systems the researchers suggested to log raw system call identifiers as data source or spatial and domain knowledge based transformation of these identifiers as features. The spatial transformation means that, the length, data values, frequency and range of data values in a system call identifiers sequence [17], whereas domain knowledge based transformations means, transforming a raw system call identifier by considering its relation with activity purpose and resource. As the traditional components of an intrusion detection system are data source, feature construction and decision engine [16]. Critical cloud servers defense based on just raw or spatial representation of system call identifiers may results in the exclusion of other useful features in the final defense mechanism; and (ii) there is a trend in hacking industry to learn the state of the art defense mechanism and then design the attacks to break them [18]. In this regard, designing and developing cyber defense systems is observed to be an ongoing process [19]. Therefore, depending on just one type of logs i.e. raw or spatial representation of system call identifiers, can minimize the reliability factor.

In this paper, the two main contributions are as follows: (i) In order to explore the new features in the theory of host based anomaly detection systems, the short sequence raw system call identifiers are transformed into frequency domain by applying sine and Fourier transformation, and (ii) To evaluate and compare the capability of proposed frequency domain feature vectors as comprehensive reflection of normal activities including discrimination power for classifying normal and attack feature vectors, different machine learning algorithms as anomaly detection engine and recently released intrusion detection system

data-set i.e. NGIDS-DS [20] are used. The considered machine learning algorithms include, SVM with linear and radial base kernels, KNN and ELM. Although anomaly intrusion detection is virtually a classical classification problem where there exist many powerful machine learning algorithms [21–23], our focus is on the construction of new features as features play a critical role. The rest of the paper is organized as follows: the literature review is given in Sect. 2; the proposed work is given in Sect. 3; experimental results and discussion are provided in Sect. 4; and the concluded remarks are given in Sect. 5.

## 2 Literature Review

In this section, the existing host based anomaly detection systems based on system calls are analyzed and classified. The classification of these systems are based on how the feature vectors are constructed by the spatial transformation or domain knowledge based manipulation of raw system calls identification. For instance, pioneer researchers of this domain utilized the raw short sequences of system call identifiers as feature vectors [24]. Later, some researchers utilized the spatial transformation of raw system call identifiers sequence i.e. considering just most frequent, less frequent, maximum and minimum system call identifiers as feature vector [16, 17, 25]. In addition some researchers have utilized the domain knowledge to manipulate raw system call identifiers in order to construct feature vectors for the host activities [26–28].

The raw short sequence of system call identifiers based host anomaly detection techniques build a model for the sub-sequences of the normal traces, and in decision engine a test occurrence opposing considerably from the model established will be reflected as abnormal. For example, in pioneer host intrusion detection works by Forests [24, 29], the feature matrix is constructed by sliding window of fixed length across the normal traces and at decision engine a trial trace comprising a percentage of mismatch away from a threshold is considered as abnormal. Tackling the long traces, Kosoresow et al. modified the look-ahead algorithms by calculating the divergences within small, fixed-length sectors of the traces [30]. Furthermore, at decision engine of the short sequence based techniques, statistical learning notions are widely adopted to predict the behavior (normal or abnormal) by summarizing the intrinsic associations concealed behind the normal traces. The example includes, artificial neural network (ANN) [31, 32], SVM [33], hidden Markov model (HMM) [34, 35] and semantic data mining [26].

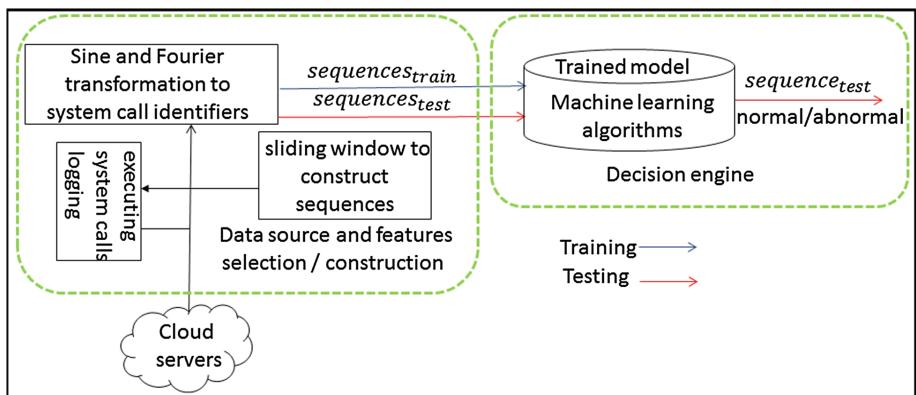
In contrast with the raw short sequence of system call identifiers as features, in [16, 17, 25] the spatial transformation of raw system call identifiers are presented to construct feature vectors for the host activities. For example, in [16, 17] a feature vector of a trace of the raw system call identifiers is constructed by just considering most frequent, less frequent, minimum, maximum and even/odd count of system call identifiers in terms of integer data. Similarly, in [25] for windows operating system, the count of key dll calls is used to construct the feature vectors of host activities. Moreover, in [26–28, 36] the domain knowledge is considered to construct feature vectors by using system call identifiers. For example,

in [27] the authors suggested the criteria for the selection of a few system calls to conduct audit, which is based on attack domain knowledge. This approach has considered only those system call identifiers, which are assumed to be involved in privileged transition flows, during attack and normal behavioral scenarios. Similarly, in [28] the traces of system call identifiers are suggested to be represented with eight kernel modules. Further, in [36], a model is presented based on system calls arguments (i.e. execution path) and sequences incorporated with clustering. The key characteristic is the consideration of different ways of using a system call in a specific process as ingredient to construct a feature vector.

In the above discussion, it can be observed that, in order to ensure the reliability of host defense by countering the current threats there is still a need to investigate the novel and hidden features and as a addition in this work we proposed the sine and Fourier transformation at the raw system call identifiers to extract frequency domain features from host operating system. To the best of our knowledge, this work can open a new way may to investigate the application of frequency domain transformation to system call identifiers in building host defense.

### 3 Proposed Work

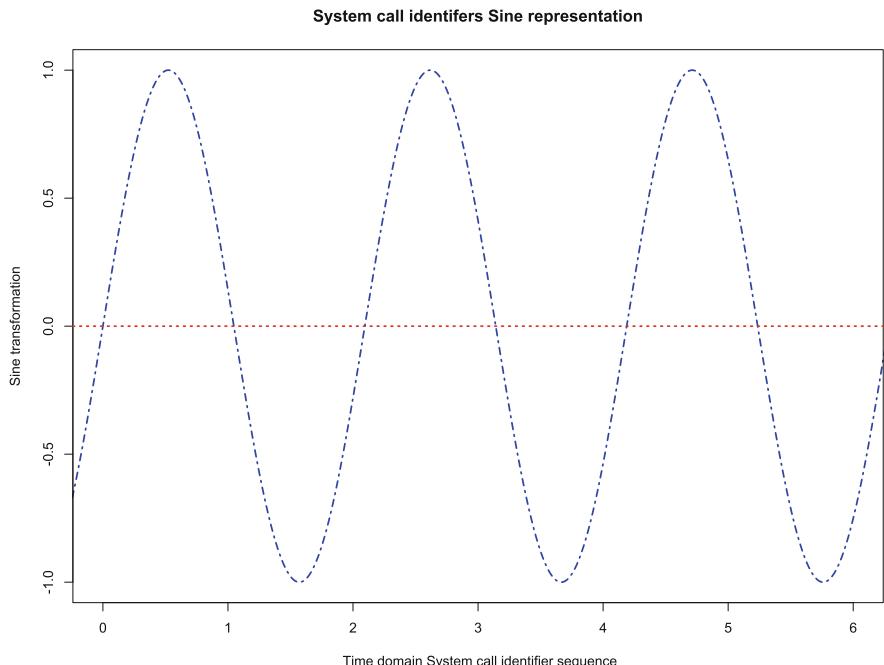
In this section, the proposed work is elaborated in terms of training and testing framework given in Fig. 1, for host based anomaly detection systems (HADS). The key contribution is the application of sine and Fourier transformation in the feature construction phase of the proposed HADS. At the feature construction phase of the existing HADSs, the spatial and domain knowledge transformations have been applied, therefore it is intended to investigate the applicability of sine and Fourier transformation as feature construction and later its impact in detecting host anomalies. Further, for performance comparison, at the decision



**Fig. 1.** Proposed HADS framework

engine of the proposed HADS different machine learning algorithms are configured independently such as SVM, KNN and ELM.

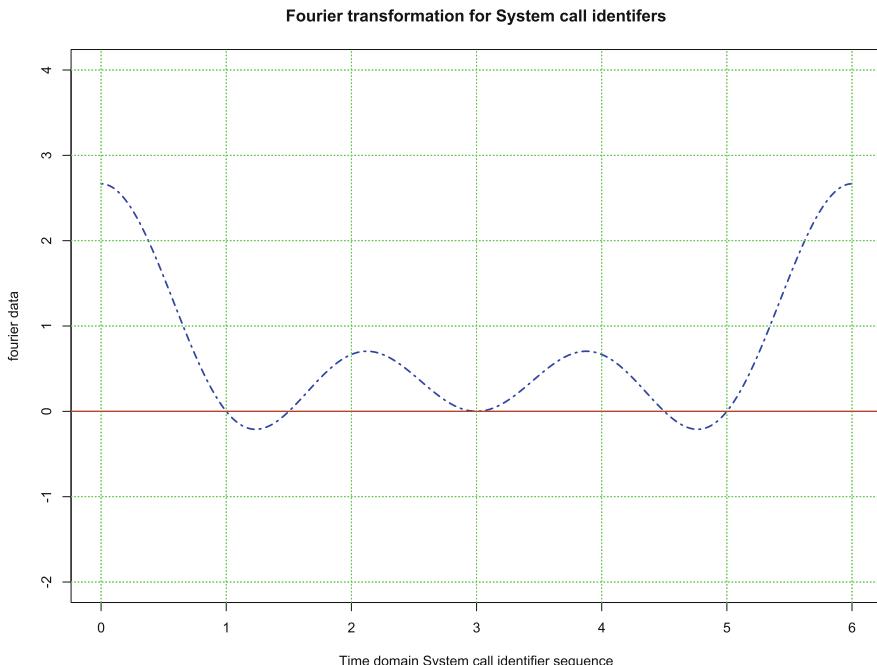
An online HADS starts its operation by first logging the comprehensive data from machines operating system. In the proposed HADS we are dealing with Linux or UNIX based operating systems where the system calls calling are considered to be the comprehensive audit data [13] that can be logged in an online manner as shown in Fig. 1. In addition most of the critical machines around the globe are comprised of these operating systems [37]. Once the data unit is logged, then a traditional HADS feature selection or construction mechanisms are triggered [38]. In the proposed HADS, we adopted first time the sine and Fourier transformation to transform the raw system call identifiers time domain signal (i.e. shown in Fig. 2) into frequency domain signal by utilizing the scheme in [39]. In order to log a unit data and to apply sine and Fourier transformation according to time, a sliding window with 1 s length is adopted i.e. each one second signal of system call identifiers is transformed in to frequency domain. Further, once the incoming system call identifier signal is transformed into the frequency domain, the feature vector is constructed. The formal description of the transformation process is elaborated as follows. First the input time domain signal of system call identifiers in 1 s is represented with sine transformation that is defined in Eq. (1). In Eq. (1), the variable  $x$  shows the system call identifier and  $t = 1$  to  $T$  and  $T$  can be 1 s.



**Fig. 2.** Time domain representation of system call identifier sequence with sine

$$f(x) = \sin(x)_t \quad (1)$$

For instance, the system call identifiers range is from 0 to 350 in considered version of the kernel of the Linux operating system (i.e. Ubuntu 14.04), however this range can be vary depending the version of the kernel. The first 6 system call identifiers sine transformation is elaborated in Fig. 2. In the frequency transformation process, after sine conversion to input signal then the Fourier transformation is applied on sine transformed signal that is defined in Eq. (2) where for any real number  $\xi$  (i.e. the sine transformed values of system call identifiers), the independent variable  $x$  represents time (with SI unit of seconds) and the transform variable  $\xi$  represents frequency (in hertz). In Fig. 3, the Fourier transformation for the first 6 system call identifiers sine transformed signal is shown. Further, in frequency transformation process, the Fourier transformed components of the input signal are treated as the sequence or feature vector of the host activity in one second. These feature vectors are further utilized to train and test the adopted machine learning algorithms as anomaly detection engines for the host. The decision engine of the proposed HADS is configured with three machine learning algorithms respectively i.e. SVM, KNN and ELM. The purpose is to evaluate the performance in terms of accuracy and error, of frequency domain feature vectors in building normal profile and to classify anomalous feature vectors. The parameters of the selected machine learning algorithms which



**Fig. 3.** Fourier transformation of system call identifiers sine representation

are empirically observed optimum are as follows. SVM (rbf) [16, 17] is configured with the parameters  $n = 10$  (cross validation value),  $s = 0$  (default type of SVM),  $d = 5$  (degree in kernel function) and rest all on default values. KNN [40] is configured with  $k = 10$  (e.g. k-fold cross validation). ELM [41] with number of hidden neurons = 50, activation function = radbas, sigmoid, sin and all data points of the feature matrix are normalized between the scale 1 and -1.

$$f(\xi) = \int_{-\infty}^{\infty} f(x)e^{-2\pi\xi x} dx \quad (2)$$

---

**Algorithm 1.** Anomaly Detection: Training and Testing

---

**Require:** system call identifiers

**Ensure:** test sequence is normal or abnormal

**Training**

- 1:Sine transformation to input data using eq(1)
- 2:Fourier transformation to data collected from step 1 and using eq(2)
- 3:Train [SVM or KNN or ELM]  $\leftarrow$  sequences from step 2

**Testing**

- 4: Repeat step 1 and 2 respectively
  - 5: Predict a sequence as normal or abnormal  $\leftarrow$  Trained [SVM or KNN or ELM]
  - 6: Go to step 4 until the last test sequence
  - 7: End
- 

In order to automate the above discussed framework of proposed HADS, Algorithm 1 is developed. In Algorithm 1, at the training, the normal or abnormal input signal (i.e., in the case of experimenting with labeled host IDS dataset) of system call identifiers are fragmented according to 1s of sliding window and transformed into the frequency domain as discussed above. The reason to adopt 1s length of sliding window is to extract frequency domain information from short sequence of system calls while short sequence of system calls are acknowledged as the good discriminator between normal and abnormal [24]. Further, the adopted machine learning algorithms are trained respectively with input frequency domain feature vectors. In Algorithm 1, at testing the trained machine learning models can predict/classify the input test frequency domain feature vector for normal or abnormal.

## 4 Experiments and Results

The proposed HADS given in Fig. 1 is evaluated using the criteria given in [16, 17]. The purpose of this section is to answer the following questions: (i) How can the accuracy of the HADS be improved by employing frequency domain transformation to system call identifiers? (ii) Is it possible to minimize the error in detecting host anomalies while adopting frequency domain feature vectors for the host activities? And (iii) what is the impact of host activities frequency domain information in building normal profile and in detecting anomalies?

In our experiment we utilized three IDS data-sets namely, ADFA-LD [42], KDD 98 [43], and NGIDS-DS [20]. ADFA-LD is a small data-set with fewer

attacks and normal data collection, whereas KDD 98 is outdated in-terms of modern attacks and normal computer activities foot prints. However, both these data-sets are utilized to compare the performance of proposed Algorithm 1. The training and testing traces of both these data-sets are acquired from [17]. Further, the modern IDS data-set (i.e., NGIDS-DS) which is generated with the maximum possible quality of realism, in the next generation cyber range infrastructure of the Australian Centre for Cyber Security (ACCS) at the Australian Defence Force Academy (ADFA), Canberra, which is designed according to the guidelines provided in [44]. The key advantage of this infrastructure is the availability of the IXIA Perfect Storm hardware. The combination of a network traffic-generation appliance and virtual cyber range provides both legitimate traffic and host-based connectivity. The IXIA Perfect Storm tool provides four major capabilities. Firstly, it can produce a mixture of modern normal and unknown abnormal cyber traffic. Secondly, it can generate the maximum number and type of zero-day attacks with different dynamic behaviors based on packs that exploit known Common Vulnerability Exposures (CVE). Thirdly, it can establish profiles of the cyber traffic of multiple enterprises. Fourthly, it can generate ground truth automatically. Moreover, the composition of all three data-sets is given in Table 1, where roughly 1:5 training to testing ratio is adopted with normal data as suggested in [38].

**Table 1.** Data-sets composition for training and testing Algorithm 1

Data-sets	Normal training data	Normal validation data	Test attack data
NGIDS-DS records	17,758,345	71,033,389	1,262,426 records
ADFA-LD traces	833	4372	746
KDD 98 traces	1076	4305	465

The accuracy and error comparison of three machine learning algorithms which are adopted in Algorithm 1 for three data-sets, is given in Table 2. According to [16, 17] DR is calculated at testing phase of the Algorithm 1 by dividing the number of detected abnormal sequences to the total number of abnormal sequences. Further, for FAR, first false positive and negative rates (i.e., FPR and FNR) are calculated at the testing phase of Algorithm 1 respectively. FPR is calculated by dividing the detected normal sequences as abnormal to the total number of normal attacks, whereas FNR is measured by dividing the number of abnormal sequences detected as normal to the total number of abnormal sequences. Lastly, the FAR is calculated as average joint error which is defined as  $\text{FAR} = \text{FNR} + \text{FPR}/2$ .

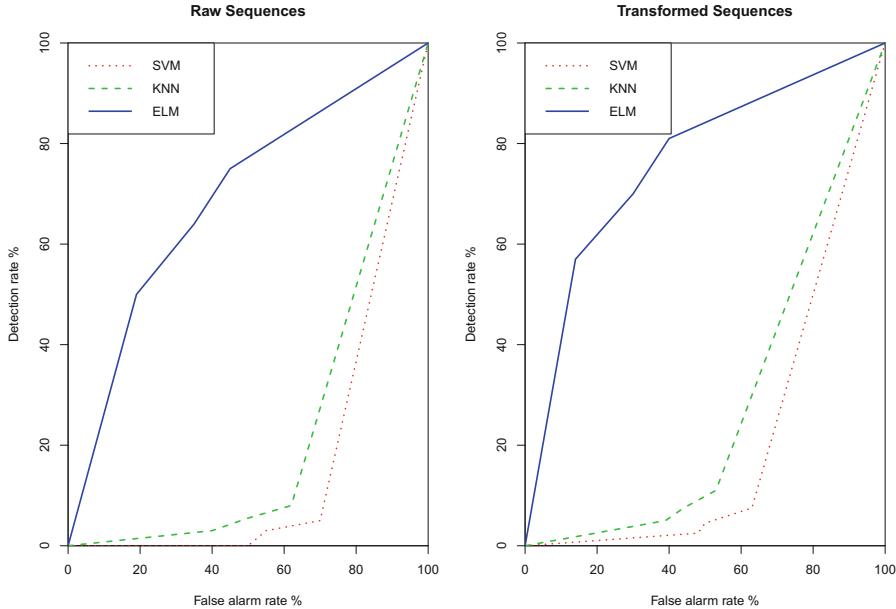
It can be observed from Table 2 that, by transforming the raw system call identifiers sequences into frequency domain have significant impact on the accuracy of proposed HADS. For instance, there is a significant increase in the DR and decrease in the FAR at each machine learning algorithm using frequency transformed sequences. The major reason for this accuracy improvement is in fact the extraction of hidden features (i.e., frequency of amplitudes in a time) from the raw system call identifiers and then the inclusion of these features as

**Table 2.** Accuracy and error comparison of system call identifiers raw and frequency domain sequences with multiple data-sets

Data-sets	Algorithms	Raw				Transformed			
		DR%	FAR%	FNR%	FPR%	DR%	FAR%	FNR%	FPR%
NGIDS-DS	SVM(rbf)	5	50	99	0.2	7.2	49	99	0.2
	KNN	8	50	99	0.6	10.5	48	99	0.5
	ELM	75	19	18	21	81	14	13	15
ADFA-LD	SVM(rbf)	70	20	30	10	75	17	26	9
	KNN	60	20	39.2	2	67	16.6	33	0.9
	ELM	88	17	12	23.7	95	11.47	5	16
KDD 98	SVM(rbf)	44	55	57	52	61	46	30	61.8
	KNN	34	68	65.1	70	48	53	52	49.7
	ELM	91	5	8.09	3	97	2	3	0.56

pre-classification assistance to machine learning algorithms. Also, all three algorithms performances are low upon NGIDS-DS data-set. The reasons behind this fact are: (i) both ADFA-LD depicts less complex data-set with small number of attacks and normal activities footprints; (ii) Kdd 98 is outdated and less complex, with inclusion of small number of high foot print attacks and differentiable normal computer activities reflection; and (iii) NGIDS-DS is complex data-set with inclusion of huge number of modern low foot print attacks and normal computer activities [13].

Further, it can be observe from Fig. 4 and Table 2 that, SVM and KNN performances are low as compared to ELM upon NGIDS-DS. The reasons for this aspect are as follows: (i) As the data-set NGIDS-DS [20] is recently released and it reflects modern sophisticated ways of conducting attacks that constitutes low foot print upon host logs i.e., system call identifier sequences of the processes. Due to this, in the data set the normal to attack records ratio is about 90:1. Hence, it is observed complex for SVM and KNN to distinguish the data points in two classes where the one class is the majority class [45, 46]; (ii) system call identifiers sequence actually represents any type of activity (e.g. legitimate or illegal) that occurred at the host but from machine learning classifier point of view it constitutes a high similarity between the data points for normal and attack sequences. Hence it is challenging for the selected SVM and KNN versions to distinguish the sequences or vectors having similar data values [45, 46]; (iii) in ELM, a single hidden layer feed-forward NN selects randomly hidden layers and determine the output weight (e.g. weight times feature vectors) for fitting the target output about any feature vector in a feature matrix. The hidden layers do not need to be tuned iteratively as compared to traditional ANN and the activation functions are adoptable [41] and (iv) in ELM, the data points of a feature vector are transformed into another domain or extended dimension using activation functions as kernels such as sigmoid, sin, and raidbas. As a result,



**Fig. 4.** Using latest IDS data-set (NGIDS-DS), raw and frequency transformed system call identifiers sequences comparison with multiple machine learning algorithms in terms of anomaly detection accuracy and error via ROC curves

ELM classifier is able to discriminate between the feature vectors of different classes by learning the natural hidden patterns which are not visible with the data points in raw domain [47].

## 5 Conclusion

It is vital to protect machines operating systems in the current and future era of cyber threats, where attacks saturation power is observed able to penetrate the network defense zones. To deal with this, an anomaly detection mechanism for cloud servers is proposed and investigated in this paper. In the proposed host based anomaly detection system, first, the audit data from LINUX/UNIX based cloud servers (i.e., system call identifiers) is transformed into frequency domain by sine and Fourier transformation from time domain, in order to extract frequency domain feature vectors of running activities at the host. Second, different machine learning algorithms are trained and tested with these frequency domain feature vectors as anomaly detection engine. Results, demonstrate that, these frequency domain features of host activities identification, are capable of detecting host anomalies with minimum error. In future, it is intended to transform the other types of audit data from machines such as CPU power and memory consumption, in order to design more reliable anomaly detection system for machines operating system.

## References

1. Pabla, I., Khalil, I., Hu, J.: Intranet security via firewalls. In: Stavroulakis, P., Stamp, M. (eds.) *Handbook of Information and Communication Security*, pp. 207–219. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-04117-4\\_11](https://doi.org/10.1007/978-3-642-04117-4_11)
2. Wang, H., Zhang, Y., Cao, J.: Access control management for ubiquitous computing. *Future Gener. Comput. Syst.* **24**(8), 870–878 (2008)
3. Moustafa, N., Slay, J.: UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: *Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6. IEEE (2015)
4. Wang, Y., Wen, S., Xiang, Y., Zhou, W.: Modeling the propagation of worms in networks: a survey. *IEEE Commun. Surv. Tutor.* **16**(2), 942–960 (2014)
5. Moustafa, N., Slay, J.: The significant features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems. In: *2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS)*, pp. 25–31. IEEE (2015)
6. Cesare, S., Xiang, Y., Zhou, W.: Malwisean effective and efficient classification system for packed and polymorphic malware. *IEEE Trans. Comput.* **62**(6), 1193–1206 (2013)
7. Rudd, E., Rozsa, A., Gunther, M., Boult, T.: A survey of stealth malware: attacks, mitigation measures, and steps toward autonomous open world solutions. *IEEE Commun. Surv. Tutor.* **19**(2), 1145–1172 (2017)
8. Moustaf, N., Slay, J.: Creating novel features to anomaly network detection using DARPA-2009 data set. In: *Proceedings of the 14th European Conference on Cyber Warfare and Security*, p. 204. Academic Conferences Limited (2015)
9. Ficco, M., Palmieri, F.: Introducing fraudulent energy consumption in cloud infrastructures: a new generation of denial-of-service attacks. *IEEE Syst. J.* **11**(2), 460–470 (2017)
10. Kumarage, H., Khalil, I., Tari, Z., Zomaya, A.: Distributed anomaly detection for industrial wireless sensor networks based on fuzzy data modelling. *J. Parallel Distrib. Comput.* **73**(6), 790–806 (2013)
11. Kumarage, H., Khalil, I., Tari, Z.: Granular evaluation of anomalies in wireless sensor networks using dynamic data partitioning with an entropy criteria. *IEEE Trans. Comput.* **64**(9), 2573–2585 (2015)
12. Alabdulatif, A., Kumarage, H., Khalil, I., Yi, X.: Privacy-preserving anomaly detection in cloud with lightweight homomorphic encryption. *J. Comput. Syst. Sci.* **90**, 28–45 (2017)
13. Haider, W., Hu, J., Xie, Y., Yu, X., Wu, Q.: Detecting anomalous behavior in cloud servers by nested arc hidden SEMI-Markov model with state summarization. *IEEE Trans. Big Data* (2017)
14. Rittinghouse, J.W., Ransome, J.F.: *Cloud Computing: Implementation, Management, and Security*. CRC Press, Boca Raton (2016)
15. Zissis, D., Lekkas, D.: Addressing cloud computing security issues. *Future Gener. Comput. Syst.* **28**(3), 583–592 (2012)
16. Haider, W., Hu, J., Xie, M.: Towards reliable data feature retrieval and decision engine in host-based anomaly detection systems. In: *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 513–517. IEEE (2015)
17. Haider, W., Hu, J., Yu, X., Xie, Y.: Integer data zero-watermark assisted system calls abstraction and normalization for host based anomaly detection systems. In: *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud)*, pp. 349–355. IEEE (2015)

18. Taddeo, M., Glorioso, L.: Ethics and Policies for Cyber Operations: A NATO Cooperative Cyber Defence Centre of Excellence Initiative, vol. 124. Springer, Cham (2017). <https://doi.org/10.1007/978-3-319-45300-2>
19. Herpig, S.: Anti-war era: the need for proactive cyber security. In: Felici, M. (ed.) CSP 2013. CCIS, vol. 182, pp. 165–176. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-41205-9\\_14](https://doi.org/10.1007/978-3-642-41205-9_14)
20. Haider, W., Hu, J., Slay, J., Turnbull, B., Xie, Y.: Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling. *J. Netw. Comput. Appl.* **87**, 185–192 (2017)
21. Toh, K.-A., Tan, G.-C.: Exploiting the relationships among several binary classifiers via data transformation. *Pattern Recogn.* **47**(3), 1509–1522 (2014)
22. Toh, K.-A.: Training a reciprocal-sigmoid classifier by feature scaling-space. *Mach. Learn.* **65**(1), 273–308 (2006)
23. Tran, Q.-L., Toh, K.-A., Srinivasan, D., Wong, K.-L., Low, S.Q.-C.: An empirical comparison of nine pattern classifiers. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **35**(5), 1079–1091 (2005)
24. Hofmeyr, S.A., Forrest, S., Somayaji, A.: Intrusion detection using sequences of system calls. *J. Comput. Secur.* **6**(3), 151–180 (1998)
25. Haider, W., Creech, G., Xie, Y., Hu, J.: Windows based data sets for evaluation of robustness of host based intrusion detection systems (IDS) to zero-day and stealth attacks. *Future Internet* **8**(3), 29 (2016)
26. Creech, G., Hu, J.: A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns. *IEEE Trans. Comput.* **63**(4), 807–819 (2014)
27. Cho, S.-B., Park, H.-J.: Efficient anomaly detection by modeling privilege flows using hidden Markov model. *Comput. Secur.* **22**(1), 45–55 (2003)
28. Murtaza, S.S., Khreich, W., Hamou-Lhadj, A., Gagnon, S.: A trace abstraction approach for host-based anomaly detection. In: IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), pp. 1–8. IEEE (2015)
29. Forrest, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A.: A sense of self for unix processes. In: Proceedings of 1996 IEEE Symposium on Security and Privacy, pp. 120–128. IEEE (1996)
30. Kosoresow, A.P., Hofmeyer, S.: Intrusion detection via system call traces. *IEEE Softw.* **14**(5), 35–42 (1997)
31. Moustafa, N., Slay, J., Creech, G.: Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. *IEEE Trans. Big Data* (2017)
32. Ghosh, A.K., Schwartzbard, A., Schatz, M.: Learning program behavior profiles for intrusion detection. In: Workshop on Intrusion Detection and Network Monitoring, vol. 51462, pp. 1–13 (1999)
33. Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. In: Barbará, D., Jajodia, S. (eds.) Applications of Data Mining in Computer Security, vol. 6, pp. 77–102. Springer, Boston (2002). [https://doi.org/10.1007/978-1-4615-0953-0\\_4](https://doi.org/10.1007/978-1-4615-0953-0_4)
34. Hoang, X., Hu, J.: An efficient hidden Markov model training scheme for anomaly intrusion detection of server applications based on system calls. In: Proceedings of 12th IEEE International Conference on Networks, (ICON 2004), vol. 2, pp. 470–474. IEEE (2004)

35. Hu, J., Yu, X., Qiu, D., Chen, H.-H.: A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection. *IEEE Netw.* **23**(1), 42–47 (2009)
36. Maggi, F., Matteucci, M., Zanero, S.: Detecting intrusions through system call sequence and argument analysis. *IEEE Trans. Dependable Secure Comput.* **7**(4), 381–395 (2010)
37. Silic, M., Back, A.: Open source software adoption: lessons from linux in munich. *IT Prof.* **19**(1), 42–47 (2017)
38. Creech, G.: Developing a high-accuracy cross platform host-based intrusion detection system capable of reliably detecting zero-day attacks. Ph.D. dissertation, University of New South Wales, Canberra, Australia (2014)
39. Bracewell, R.N., Bracewell, R.N.: *The Fourier Transform and Its Applications*, vol. 31999. McGraw-Hill, New York (1986)
40. Moustafa, N., Creech, G., Slay, J.: Big data analytics for intrusion detection system: statistical decision-making using finite dirichlet mixture models. In: Palomares Carrascosa, I., Kalutarage, H.K., Huang, Y. (eds.) *Data Analytics and Decision Support for Cybersecurity*. DA, pp. 127–156. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-59439-2\\_5](https://doi.org/10.1007/978-3-319-59439-2_5)
41. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: theory and applications. *Neurocomputing* **70**(1), 489–501 (2006)
42. Creech, G., Hu, J.: Generation of a new IDS test dataset: time to retire the KDD collection. In: 2013 IEEE Wireless Communications and Networking Conference (WCNC), pp. 4487–4492. IEEE (2013)
43. KDD98 (1988). <http://www.ll.mit.edu/mission/communications/>
44. Davis, J., Magrath, S.: A survey of cyber ranges and testbeds. Defence Science and Technology Organisation Edinburgh (Australia) Cyber and Electronic Warfare Division, Technical report (2013)
45. Xing, Z., Pei, J., Keogh, E.: A brief survey on sequence classification. *ACM SIGKDD Explor. Newsl.* **12**(1), 40–48 (2010)
46. Justino, E.J., Bortolozzi, F., Sabourin, R.: A comparison of SVM and HMM classifiers in the off-line signature verification. *Pattern Recogn. Lett.* **26**(9), 1377–1385 (2005)
47. Vong, C.-M., Ip, W.-F., Wong, P.-K., Chiu, C.-C.: Predicting minority class for suspended particulate matters level by extreme learning machine. *Neurocomputing* **128**, 136–144 (2014)



# A Variant of BLS Signature Scheme with Tight Security Reduction

Tiong-Sik Ng<sup>1</sup>(✉) , Syh-Yuan Tan<sup>1</sup> , and Ji-Jian Chin<sup>2</sup>

<sup>1</sup> Faculty of Information Science and Technology,  
Multimedia University, Melaka, Malaysia

[ng.tiong.sik@gmail.com](mailto:ng.tiong.sik@gmail.com), [sytan@mmu.edu.my](mailto:sytan@mmu.edu.my)

<sup>2</sup> Faculty of Engineering, Multimedia University, Cyberjaya, Malaysia  
[jjchin@mmu.edu.my](mailto:jjchin@mmu.edu.my)

**Abstract.** In 2001, Boneh, Lynn and Shacham designed a signature scheme using the properties of bilinear pairing from elliptic curve, and based its security under the Computational Diffie-Hellman (CDH) assumption. However, the security reduction is not tight as there is a loss of roughly  $q_s$ , the number of sign queries. In this paper, we propose a variant of the BLS signature with tight security reduction based on the co-CDH assumption. Besides upgraded to the notion of strong existential unforgeability under chosen message attack, the variant is backward-compatible with the original BLS signature.

**Keywords:** BLS · Signature · Tight security reduction  
Provable security

## 1 Introduction

In cryptography, a scheme is deemed secure if its security can be proven mathematically. The property of provable security was first proposed by Goldwasser and Micali in [17]. Although a cryptography primitive can be proven secure, the security reduction in the security proof may not be tight. A scheme is also said to have a tight security if breaking the scheme is as hard as solving the assumption that the scheme uses. Therefore, a tight security reduction can achieve the same security level without using larger key size. For example, the probability of breaking BLS signature is known [5] to be approximately  $2e \cdot (q_s \times \varepsilon_{CDH})$  where  $e$  is the natural logarithm and  $q_s$  is the number of signatures an attacker can obtain, while  $\varepsilon_{CDH}$  is the probability of breaking the Computational Diffie-Hellman (CDH) problem. If we allow  $q_s = 2^{30}$  queries, initializing BLS signature scheme with BN curve of 256 bits key size which contributes to 128 bits security, the real security of BLS is only 96 bits:  $2^1 \times 2^1 \times 2^{30} \times 2^{-128} = 2^{-96}$ . The non-tight security reduction shows that the BLS signature scheme has to use a larger key size, to achieve the same 128 bits security level.

The BLS is one among two well-known signature schemes that uses the shortest signature length to date, alongside the Boneh-Boyen (BB) signature

scheme [1]. The BB signature scheme is said to have a signature length as short as the BLS signature, and is also more efficient. The scheme was designed so that the bilinear pairing only needs to be done once during the verification, as compared to the BLS scheme that needs two bilinear pairings. Apart from that, the security was also proven without the help of the random oracle. However, it is known that the security of the Strong Diffie-Hellman (SDH) problem that the BB scheme uses is approximately 40% weaker compared to the CDH problem with the same parameter length, despite not using the random oracle for the security reduction [18]. Table 1 shows the comparison of ideal parameters for the BLS signature and some well-known signature schemes at 128 bits security level with their respective security tightness. Based on the table, it can be noticed that the security tightness of the RSA-PSS [8] scheme is the only one which does not contradict with its parameter size. For the signature schemes such as the BLS and BB which require bilinear pairing, we calculate the public key and signature sizes of the schemes based on the BN curve [27], where Type 3 pairing is used. Throughout this paper, point compression would be used to represent the public keys and signature lengths for schemes that are using the BN curve.

**Table 1.** Comparison among digital signatures at 128 bit security level

Scheme	Public key size (bits)	Signature size (bits)	Security tightness
DSA [14] <sup>#</sup>	$2 \times 3072$	$2 \times 256$	$\varepsilon_{DSA} \approx \varepsilon_{DL}^*$
EC-DSA [14] <sup>#</sup>	$2 \times 256$	$2 \times 256$	$\varepsilon_{EC-DSA} \approx \varepsilon_{DL}^*$
EC-DSA+ [19]	$2 \times 256$	$2 \times 256$	$\varepsilon_{EC-DSA+} \approx (\frac{\varepsilon_{DL}}{2q_h})^3$
Schnorr [29]	$2 \times 3072$	$2 \times 256$	$\varepsilon_{Schnorr} \approx (\frac{\varepsilon_{DL}}{2q_h})^3$
EC-Schnorr [20]	$2 \times 256$	$2 \times 256$	$\varepsilon_{EC-Schnorr} \approx 6q_h \varepsilon_{DL}$
RSA-FDH [7] <sup>#</sup>	$2 \times 3072$	3072	$\varepsilon_{FDH} = (q_s + q_h + 1)\varepsilon_{RSA}$
RSA-PSS [8] <sup>#</sup>	$2 \times 3072$	3072	$\varepsilon_{PSS} = \varepsilon_{RSA}$
BLS [22]	$2 \times 256 + 2 \times 512$	256	$\varepsilon_{BLS} = e(q_s + 1)\varepsilon_{co-CDH}$
BNN-BLS [6]	$2 \times 256$	257	$\varepsilon_{BLS} = 2\varepsilon_{co-CDH}$
BB [1]	$256 + 2 \times 512 + 3072$	256	$\varepsilon_{BB} = \varepsilon_{SDH}$

\* There are no proofs for these signature schemes in the random oracle model, however it is commonly believed that the probability of breaking these schemes are as hard as breaking the discrete logarithm (DL) problem.

<sup>#</sup> Key size is following recommendation from NIST [26].

## 1.1 Related Works

After the concept of digital signatures was first proposed in [13], many digital signature schemes have emerged. Among the de-facto signature schemes are the Digital Signature Algorithm (DSA) [14] and the Schnorr signature [29]. The DSA was described based on the adoption of the ElGamal [15] signature. The DSA is a popular signature scheme that is used as a Federal Information Processing Standard, and is widely used in computer systems by non-military government

organizations. A variant of the DSA which uses the elliptic curve, also known as the EC-DSA was first proposed in [14]. The size of the EC-DSA's public key is said to be shorter than that of the DSA, while the signature length remains the same. The Schnorr signature is another signature scheme that is based on the ElGamal signature. It was first proposed to be suitable for interactions between smart cards and terminals, as the algorithm is said to be efficient.

In [5], Boneh et al. proposed the Boneh-Lynn-Shacham (BLS) digital signature scheme based on the assumption that the Computational Diffie-Hellman (CDH) problem is intractable. As stated in their work, the signature of the BLS is proposed to be only 160 bits long compared to the RSA [28] that is 1024 bits long and the DSA that is 320 bits long, while maintaining the same security levels of the latter. Throughout the years after the first appearance of the BLS signature, many variants of the BLS have appeared. Today, the BLS signature is used for various purposes such as cloud storage [30], aggregate signatures [23], and also big data [24].

In [9], Cha and Hee Cheon designed a variant of the BLS, namely, identity-based BLS signature where the user's public ID was used as a public key for verification. In [31], a ring signature scheme was designed by Zhang et al. which is very similar to a combination of the Cha-Cheon's scheme and the Boneh-Boyen's scheme. However, the scheme was not tightly secure as well. In [22], Lacharité proposed a Type-3 Pairing version of the BLS signature based on the Computational co-Diffie-Hellman Assumption (co-CDH) assumption. Her works were heavily based on Chatterjee et al.'s [10] work, where the modified co-CDH (co-CDH\*) assumption was proposed.

In [16], a signature scheme very similar to the BLS was proposed by Goh and Jarecki based on the CDH assumption. Inspired by [16], Katz and Wang [21] tightened the security of FDH signatures by hashing just one bit extra together with the message. Based on the security reduction, it is shown that the scheme is almost as secure as the CDH assumption such that  $\varepsilon_{FDH} = 2 \cdot \varepsilon_{CDH}$ . In [6], Bellare et al. proposed an aggregate version of the BLS signature using Katz-Wang's technique where the security reduction relies on the co-CDH assumption. The proposed BLS variant has a tight security reduction, which is similar to the result that Katz-Wang produced.

## 1.2 Our Contribution

In [12], Coron proposed a tight security patch for the Boneh-Franklin IBE (BF-IBE) [3] that is backward compatible based on the D-Square-BDH assumption. The upgraded BF-IBE is tightly secure with the help of a random salt drawn from the space of  $\mathbb{Z}_q$ . Inspired by Coron's work, we propose a tight security upgrade to the BLS signature, and different from Coron's technique, we only require the salt to be 1 bit in length. Moreover, the BLS variant is upgraded to a stronger security notion, namely, strong existential unforgeability under chosen message attacks (*seuf-cma*) based on the co-CDH assumption. Though our technique also uses a 1 bit salt, the salt is not hashed with  $m$  as in Katz-Wang's technique [21]. Instead, it is used as an exponent for the extra public

key element. A comparison of the original BLS signature and our variant at 128 bit security level is shown in Table 2 below.

**Table 2.** Comparison between the original BLS and our variant

Scheme	BLS (Type-1) [5]	BLS (Type-3) [22]	BNN-BLS [6]	Our variant (Type-3)
Assumption	CDH	co-CDH	co-CDH	co-CDH
Pairing type	1	3	2	3
Public key elements	$2 \mathbb{G} $	$2 \mathbb{G}_1  + 2 \mathbb{G}_2 $	$2 \mathbb{G}_1  + 2 \mathbb{G}_2 $	$3 \mathbb{G}_1  + 2 \mathbb{G}_2 $
Signature length	$ \mathbb{G} $	$ \mathbb{G}_1 $	$ \mathbb{G}_1  + 1$ bit	$ \mathbb{G}_1  + 1$ bit
Security model	<i>euf-cma</i>	<i>euf-cma</i>	<i>seuf-cma</i>	<i>seuf-cma</i>
Security tightness	$\epsilon_{BLS} = 2e \cdot q_S \epsilon_{CDH}$	$\epsilon_{BLS} = e(q_s + 1)\epsilon_{co-CDH}$	$\epsilon_{BNN-BLS} = 2\epsilon_{co-CDH}$	$\epsilon_{BLS} = \epsilon_{co-CDH}$
Backward compatibility with original BLS	Original	Original	No	Yes

### 1.3 Organization

The paper is organized as such. In Sect. 2, the definitions and security model of a digital signature will be described. In Sect. 3, the original BLS signature scheme will be described in detail. Besides that, our variant will also be described alongside its security proof. In Sect. 4, the design of the variant will be discussed in detail. We conclude our findings in Sect. 5.

## 2 Definitions

In this section, we briefly describe the definitions and the backgrounds of digital signatures and related mathematical assumptions. Throughout this paper, we let  $\{0, 1\}^*$  denote the set of all bit strings while  $\{0, 1\}^n$  the set of bit strings of length  $n$ . If a string  $s \in \{0, 1\}^*$  then  $|s|$  denotes the length of  $s$ . If  $S$  is a set then  $|S|$  denotes the size of  $S$ . Let  $a \xleftarrow{R} S$  denote a randomly and uniformly chosen element  $a$  from a finite set  $S$ . Lastly let  $\mathbb{Z}_p$  denote the set of positive integers modulo a large prime  $p$ .

**Definition 1.** *A digital signature consists of three polynomial-time algorithms: **Key Generation**, **Sign**, and **Verify**. The first two algorithms are probabilistic. The algorithms are described as follows:*

1. **Key Generation** ( $1^k$ ): A pair of public and secret keys are generated based on the security parameter input  $1^k$ . The public key  $pk$  can be aired on an open channel, while the secret key  $sk$  is kept secret by the user.
2. **Sign** ( $m, sk$ ): The user uses the secret key  $sk$  to sign on a message  $m$  to generate a signature, which is denoted as  $\sigma$ .
3. **Verify** ( $m, \sigma, pk$ ): The verifier takes the public key  $pk$  and  $\sigma$  as the input to ensure that the signature is genuinely signed by the user. If the signature is authentic, the algorithm returns “True”, and “False” otherwise.

## 2.1 Security Notions

We refer to two security notions, the *existential unforgeability under chosen message attacks (euf-cma)* and *strong existential unforgeability under chosen message attacks (seuf-cma)*. The security model of a digital signature is defined as the following:

1. **Setup.** During this stage, the Simulator  $\mathcal{S}$  generates and passes the public parameters to Adversary  $\mathcal{A}$ .
2. **Hash Query.**  $\mathcal{A}$  is allowed to make multiple hash queries on a message  $m$  in order to obtain  $H(m)$ .
3. **Sign Query.**  $\mathcal{A}$  is allowed to make multiple signature queries on a message  $m$  in order to obtain  $\sigma$ .  $\mathcal{S}$  would compute and send  $\sigma$  to  $\mathcal{A}$ .
4. **Forgery.** After obtaining sufficient information,  $\mathcal{A}$  would output a message and signature pair,  $(m^*, \sigma^*)$ , where  $m^*$  is a message that has not been signed before if it is a *euf-cma*  $\mathcal{A}$ ; else  $m^*$  is signed before but  $\sigma^*$  is not the previously returned signature if it is a *seuf-cma*  $\mathcal{A}$ . The forgery is successful if  $(m^*, \sigma^*)$  is a valid message-signature pair.

**Definition 2.** A digital signature scheme is  $(t, q_h, q_s, \varepsilon)$ -secure against existential forgery under adaptive chosen message attacks (*euf-cma*) if for any adversary  $\mathcal{A}$  who runs in time  $t$  succeeds in forging a signature for a message that has not been signed before, i.e.

$$|\Pr[\text{Ver}(pk, m^*, \sigma^*) = 1 : (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{sk(\cdot)}}(pk); (m^*, *) \notin \mathcal{Q}]| \leq \text{negl}(n).$$

where  $\mathcal{A}$  can make at most  $q_h$  hash queries and  $q_s$  signing queries.

**Definition 3.** A digital signature scheme is  $(t, q_h, q_s, \varepsilon)$ -secure against strong existential forgery under adaptive chosen message attacks (*seuf-cma*) if for any adversary  $\mathcal{A}$  who runs in time  $t$  succeeds in forging a signature for a message that has previously been signed before, i.e.

$$|\Pr[\text{Ver}(pk, m, \sigma^*) = 1 : (m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{sk(\cdot)}}(pk); (m, \sigma^*) \notin \mathcal{Q}]| \leq \text{negl}(n).$$

where  $\mathcal{A}$  can make at most  $q_h$  hash queries and  $q_s$  signing queries.

## 2.2 Bilinear Pairing

Let  $\mathbb{G}_1$  and  $\mathbb{G}_2$  be groups of prime order  $q$  based on the curve  $E$  over the finite field  $\mathbb{F}_p$  where  $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ . Let  $g_1$  be a generator of  $\mathbb{G}_1$  and  $g_2$  be a generator of  $\mathbb{G}_2$ . Bilinear pairing is a function which maps elements from group  $\mathbb{G}_1$  and group  $\mathbb{G}_2$  to group  $\mathbb{G}_T$ , i.e.  $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ . The bilinear pairing function  $e$  requires the following properties:

1. Bilinearity:  $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ .
2. Non-degeneracy:  $e(g_1, g_2) \neq 1$
3.  $e$  is efficiently computable, which means there is an algorithm to compute  $e(g_1, g_2)$  for any  $g_1 \in \mathbb{G}_1$  and  $g_2 \in \mathbb{G}_2$ .

## 2.3 Computational Assumptions

We adopt the definition of the CDH assumption from [2] as follows:

**Definition 4.** *Computational Diffie-Hellman (CDH) Assumption.* An algorithm  $\mathcal{S}$  is said to  $(t, \varepsilon)$ -solve the CDH problem if  $\mathcal{S}$  runs in time at most  $t$  and furthermore:

$$|\Pr[a, b \leftarrow \mathbb{Z}_q : \mathcal{S}(g, g^a, g^b) = g^{ab}]| \geq \varepsilon$$

We say that the CDH assumption is  $(t, \varepsilon)$ -hard if no algorithm  $(t, \varepsilon)$ -solves the CDH assumption.

We adopt the definition of the co-CDH assumption<sup>1</sup> from [10] as follows:

**Definition 5.** *Computational co-Diffie-Hellman (co-CDH) Assumption.* An algorithm  $\mathcal{S}$  is said to  $(t, \varepsilon)$ -solve the co-CDH problem if  $\mathcal{S}$  runs in time at most  $t$  and furthermore:

$$|\Pr[a, b \leftarrow \mathbb{Z}_q : \mathcal{S}(g_1, g_1^a, g_1^b, g_2^a) = g_1^{ab}]| \geq \varepsilon$$

We say that the co-CDH assumption is  $(t, \varepsilon)$ -hard if no algorithm  $(t, \varepsilon)$ -solves the co-CDH assumption.

*Note:* The relationship between the co-CDH assumption and the CDH assumption is not studied in Chatterjee et al.'s work [10]. Therefore, it is not known if there are any security gaps between the CDH assumption and the co-CDH assumption. However, it can be said that the co-CDH assumption is a Type-3 Pairing version of the CDH assumption which uses the Type-1 Pairing [6].

---

<sup>1</sup> The co-CDH assumption was first proposed by Boneh et al. in [4]. Our scheme leans towards the modified co-CDH (co-CDH\*) assumption proposed by Chatterjee et al. in [10]. However, we use the co-CDH assumption throughout this paper for simplicity, as the co-CDH and co-CDH\* assumptions are equivalent [10].

## 2.4 Pseudorandom Bit Generator

A pseudorandom bit generator is an efficiently computable function. Given an output sequence of the generator  $F^1$  and a truly random sequence of the same length  $F^2$ , a distinguishing algorithm  $\mathcal{S}$  cannot correctly distinguish the function with a probability of more than  $1/2$ , i.e.

$$\text{Adv}_{\mathcal{S}, \text{PRBG}}^{\text{prbg}}(n) = \left| \Pr_{g \xleftarrow{R} F^2} [\mathcal{S}^g = 1] - \Pr_{g \xleftarrow{R} F^1} [\mathcal{S}^g = 1] \right| = \frac{1}{2} + \varepsilon$$

where probabilities are over the choices of  $g$  and the coin tosses  $\mathcal{S}$  for non negligible  $\varepsilon$  (e.g.  $\varepsilon = 1/1000$ ).

## 3 The New BLS Signature Scheme

Before presenting our upgrade to the BLS signature scheme, we first recall the original BLS signature scheme in the Type-3 pairing setting.

### 3.1 The BLS Signature Scheme

The BLS signature scheme [22] is defined as follows:

1. **Key Generation:** Choose generators  $g_1 \xleftarrow{R} \mathbb{G}_1$  and  $g_2 \xleftarrow{R} \mathbb{G}_2$ , and generate a random integer  $a \xleftarrow{R} \mathbb{Z}_q^*$ . Then set  $x_1 = g_1^a$  and  $y = g_2^a$  as well as select a hash function  $H : \{0,1\}^* \rightarrow \mathbb{G}_1$ . Lastly establish the pairing function  $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ . Publish the public keys as  $\{g_1, g_2, x_1, y, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, H\}$  and keep  $a$  as the secret key.
2. **Sign:** Given a message  $m$  and secret key  $a$  as input, compute the signature as  $\sigma = H(m)^a$ .
3. **Verify:** To verify the signature  $\sigma$  of a message  $m$ , check the validity of the tuple  $(H(m), y, \sigma, g_2)$  by resolving  $e(H(m), y) = e(\sigma, g_2)$ .

For correctness, the following equation should hold:

$$\begin{aligned} e(H(m), y) &= e(H(m), g_2^a) \\ &= e(H(m)^a, g_2) \\ &= e(\sigma, g_2) \end{aligned}$$

*Note:* The public key  $x_1$  is not used throughout the scheme, but it is used for the security proof in [22].

### 3.2 The New Construction

In order to improve the original BLS scheme, we require the addition of a bit  $r \in \{0, 1\}$  and a new secret key  $b \xleftarrow{R} \mathbb{Z}_q^*$ . The new construction is described in detail as the following:

1. **Key Generation:** Choose generators  $g_1 \xleftarrow{R} \mathbb{G}_1$  and  $g_2 \xleftarrow{R} \mathbb{G}_2$ , and generate random integers  $a \xleftarrow{R} \mathbb{Z}_q^*$  and  $b \xleftarrow{R} \mathbb{Z}_q^*$ . Then set  $x_1 = g_1^a$ ,  $x_2 = g_1^b$  and  $y = g_2^a$  as well as select a hash function  $H : \{0, 1\}^* \rightarrow \mathbb{G}_1$  and pseudorandom bit generator  $PRBG : \{0, 1\}^* \times \mathbb{Z}_q^* \times \mathbb{Z}_q^* \rightarrow \{0, 1\}$ . Lastly establish the pairing function  $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ . Publish the public keys as  $\{g_1, g_2, x_1, x_2, y, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, H, PRBG\}$  and keep  $\{a, b\}$  as the secret keys.
2. **Sign:** Given a message  $m$  and secret keys  $\{a, b\}$  as input, generate a bit  $r \leftarrow PRBG(m, a, b)$  and compute  $(H(m) \cdot x_2^{-r})^a$ . The signature is generated as  $\sigma = (\delta, r) = ((H(m) \cdot x_2^{-r})^a, r)$ .
3. **Verify:** To verify the signature  $\sigma = ((H(m) \cdot x_2^{-r})^a, r)$  of a message  $m$ , check the validity of the tuple  $(\delta, g_2, H(m) \cdot x_2^{-r}, y)$  by resolving  $e(H(m) \cdot x_2^{-r}, y) = e(\delta, g_2)$ .

For correctness, the following equation should hold:

$$\begin{aligned} e(H(m) \cdot x_2^{-r}, y) &= e(H(m) \cdot x_2^{-r}, g_2^a) \\ &= e((H(m) \cdot x_2^{-r})^a, g_2) \\ &= e(\delta, g_2) \end{aligned}$$

*Note:* Similar to the original BLS scheme using Type 3 pairing in [22], the public key  $x_1$  is not used throughout the scheme. However, it is used in our security proof.

Our scheme can be used as an upgrade on the original BLS scheme, as the signing and verification algorithms are backward compatible with that of the original BLS signature algorithms. Particularly, the original signing algorithm only needs to multiply the signature  $H(m)^a$  with  $x_2^{-ra}$ , while the original verification algorithm multiplies the left handside  $e(H(m), y)$  with  $e(x_2^{-r}, y)$ . The difference with the original BLS scheme is that a “randomization” of a bit  $r \leftarrow PRBG(m, a, b)$  was added<sup>2</sup> to the signature<sup>3</sup>. However, as the bit  $r$  is chosen during the generation of each signature, there is only one valid bit  $r$  for a given message<sup>4</sup>.

---

<sup>2</sup> We propose the usage of a single bit similar to Katz-Wang’s technique in [21] to optimize the signature length. However, the security proof for an integer instead of a bit  $r$  works just as well as the RSA-PFDH [11]. The security of PRBG to randomize the signature is not an issue, as proposed and used by Katz-Wang [21] and Koblitz-Menezes [19].

<sup>3</sup> To avoid having a state where two signatures for a message exist at once where the value of the bit  $r$  may be either 0 or 1, the signer may enclose the bit  $r$  alongside  $\sigma$  to avoid further confusion during verification.

<sup>4</sup> The value of  $r$  cannot be changed as once the signature is generated, the value of  $\delta$  in the signature would be corrupted if the value of  $r$  is of a different value.

### 3.3 Security Proof

**Theorem 1.** *The new BLS signature is  $(t, q_h, q_s, \varepsilon)$ -seuf-cma secure if the co-CDH assumption is  $(t', \varepsilon')$ -hard, where:*

$$\begin{aligned}\varepsilon &= \varepsilon' \\ t &= \mathcal{O}(t')\end{aligned}$$

*Proof.* Assume that there exists a  $(t, q_h, q_s, \varepsilon)$ -adversary  $\mathcal{A}$  running in time of at most  $t$  making at most  $q_h$  hash queries and at most  $q_s$  signing queries against the new BLS scheme which forges a valid signature with probability of at least  $\varepsilon$ . We construct a simulator  $\mathcal{S}$  that solves the co-CDH problem with an advantage of at least  $\varepsilon'$  while interacting with  $\mathcal{A}$ .

**Setup.**  $\mathcal{S}$  receives the co-CDH challenge  $\{g_1, g_1^a, g_1^b, g_2^a\}$  and must output  $g_1^{ab}$ .  $\mathcal{S}$  sets  $g_1 = g_1$ ,  $x_1 = g_1^a$ ,  $x_2 = g_1^b$  and  $y = g_2^a$ . The master keys  $\{a, b\}$  are not known to  $\mathcal{S}$ .

**Hash Query.** When  $\mathcal{A}$  submits<sup>5</sup> a fresh query  $H(m)$  for message  $m$ ,  $\mathcal{S}$  generates random values  $p_1, p_2, p \xleftarrow{R} \mathbb{Z}_q^*$ , and then computes a random bit  $\tilde{r} \leftarrow PRBG(m, p_1, p_2)$ .  $\mathcal{S}$  then stores  $\{m, p, \tilde{r}\}$  in  $H$ -list and returns  $H(m) = g_1^p \cdot x_2^{\tilde{r}}$ . If  $m$  was queried before,  $\mathcal{S}$  searches for the existing record from  $H$ -list and returns the same  $H(m) = g_1^p \cdot x_2^{\tilde{r}}$ .

**Sign Query.** When  $\mathcal{A}$  submits a signing query for  $m$ , we assume the hash query  $H(m)$  has already been made. If not,  $\mathcal{S}$  goes ahead and computes the hash query first. In either case,  $\mathcal{S}$  can recover  $(p, \tilde{r})$  from  $H$ -list and let  $\delta = x_1^p$  to return  $\sigma = (\delta, \tilde{r})$ . This is a valid signature for  $m$  as:

$$\begin{aligned}\delta &= (H(m) \cdot x_2^{-\tilde{r}})^a \\ &= (g_1^p \cdot x_2^{\tilde{r}} \cdot x_2^{-\tilde{r}})^a \\ &= g_1^{ap} \\ &= x_1^p\end{aligned}$$

It should be noted that  $\mathcal{S}$  can always answer the signature queries made by  $\mathcal{A}$ .

**Forgery.** Without loss of generality, we assume the message  $m^*$  used in the forgery  $(m^*, \sigma^* = (\delta^*, r^*))$  was queried to hash oracle. If that is not the case,  $\mathcal{S}$  issues a hash query for  $m^*$ . We distinguish the forgery of  $\mathcal{A}$  into 2 cases:

*Case 1:* Suppose  $\mathcal{A}$  produces a valid  $(m^*, \sigma^* = (\delta^*, r^*))$  pair where the signature of  $m^*$  is never queried by  $\mathcal{A}$  before<sup>6</sup>,  $\mathcal{S}$  aborts if  $r^* = \tilde{r}$ ; if  $r^* \neq \tilde{r}$ ,  $\mathcal{S}$  goes ahead to solve the co-CDH assumption.

<sup>5</sup> Different from Katz-Wang's work in [21],  $\mathcal{A}$  is not allowed to query the value of  $r$ , since it is not part of the hash inputs.

<sup>6</sup> In this case,  $\mathcal{A}$  falls under the category of an *euf-cma* Adversary, whose  $m^*$  in the forgery must not be signed before.

*Case 2:* Suppose  $\mathcal{A}$  produces a valid  $(m^*, \sigma^* = (\delta^*, r^*))$  pair where  $\sigma^*$  is not the response given by  $\mathcal{S}$  during the sign query<sup>7</sup>,  $\mathcal{S}$  goes ahead to solve the co-CDH assumption.

When  $r^* \neq \tilde{r}$ ,  $\mathcal{S}$  can solve the co-CDH assumption by extracting  $g_1^{ab}$  as follows:

$$\begin{aligned} \left( \frac{\delta^*}{x_1^p} \right)^{\frac{1}{(\tilde{r}-r^*)}} &= \left( \frac{(H(m^*) \cdot x_2^{-r^*})^a}{g_1^{ap}} \right)^{\frac{1}{(\tilde{r}-r^*)}} \\ &= \left( \frac{((g_1^p \cdot x_2^{\tilde{r}}) \cdot x_2^{-r^*})^a}{g_1^{ap}} \right)^{\frac{1}{(\tilde{r}-r^*)}} \\ &= \left( \frac{((g_1^p \cdot g_1^{b\tilde{r}})g_1^{-br^*})^a}{g_1^{ap}} \right)^{\frac{1}{(\tilde{r}-r^*)}} \\ &= \left( \frac{(g_1^{ap})(g_1^{ab(\tilde{r}-r^*))}}{g_1^{ap}} \right)^{\frac{1}{(\tilde{r}-r^*)}} \\ &= g_1^{ab} \end{aligned}$$

Since  $\mathcal{S}$  can answer all hash and sign queries in either case, the probability of breaking the co-CDH assumption is:

$$\Pr[\mathcal{S} \text{ solves co-CDH}] = \Pr[\mathcal{A} \text{ outputs valid } \sigma^* \wedge \mathcal{S} \text{ does not abort}]$$

$$\varepsilon' = \Pr[\mathcal{A} \text{ outputs valid } \sigma^*] \Pr[\mathcal{S} \text{ does not abort hash queries}]$$

$$\Pr[\mathcal{S} \text{ does not abort sign queries}] \Pr[r^* \neq \tilde{r}]$$

$$\varepsilon' = \varepsilon \times 1 \times 1 \times 1$$

$$\varepsilon' = \varepsilon$$

Recall that  $\mathcal{A}$  is a forger in the security notion of *seuf-cma*. If  $m^*$  was previously queried to the sign oracle, the returned signature would be  $\sigma = (\delta, \tilde{r})$ , and so the forged signature  $\sigma^*$ , which is different from  $\sigma$ , has to be  $\sigma^* = (\delta^*, r^*)$  such that  $r^* \neq \tilde{r}$ . Since  $r \in \{0, 1\}$ ,  $r^* \neq \tilde{r}$  happens with probability 1 and subsequently  $\delta^* \neq \delta$ . On the other hand, in Case 1, which is the security notion of *euf-cma*,  $\mathcal{A}$  does not query  $m^*$  to sign oracle before, and  $\Pr[r^* \neq \tilde{r}] = 1/2$  happens on the forged signature  $\sigma^*$ . Since we are considering an upgrade to the original BLS signature scheme, we emphasize Case 2 only, which is the *seuf-cma* notion as stated in Theorem 1. The time needed to break the scheme,  $t_{\mathcal{A}}$  is defined as the computation time throughout the scheme,  $\mathcal{O}(t)$  and  $\varepsilon' = \varepsilon$  as expected.  $\square$

## 4 Discussion

### 4.1 Public Keys and Signature Length

To achieve a 128 bit security, the ideal size of a public key for the BLS signature on BN curve would be  $2|\mathbb{G}_1| + 2|\mathbb{G}_2| = 2(|\mathbb{G}_T|/12 \times 2) + 2(|\mathbb{G}_T|/6 \times 2) =$

---

<sup>7</sup> In this case,  $\mathcal{A}$  falls under the category of a *seuf-cma* Adversary, whose  $m^*$  in the forgery must be signed before.

$2(3072/12 \times 2) + 2(3072/6 \times 2) = 2(256 \times 2) + 2(512 \times 2) = 3072$  bits, or 1536 bits if point compression is used. However, as the original BLS scheme is not tightly secure, a larger public key of  $(2(7680/12 \times 2) + 2(7680/6 \times 2))/2 = (2(640 \times 2) + 2(1280 \times 2))/2 = 7680/2 = 3840$  bits is needed to make up for the loss to achieve the same 128 bits security. A comparison of the key pairs and signature lengths between the original BLS scheme and our variant is shown in Table 3.

**Table 3.** Public key length and signature length

Scheme	BLS (Type-3) (ideal)	BLS (Type-3) [22]	BNN-BLS (Type-3) [6]	Our variant
Public key length	1536 bits	3840 bits	1536 bits	1792 bits
Secret key length	256 bits	384 bits	256 bits	512 bits
Signature length	256 bits	640 bits	257 bits	257 bits
Backward compatibility	Original	Original	No	Yes

Although our upgrade adds an extra  $\mathbb{G}_1$  element to the public key, we still manage to reduce the size of the public key from 3840 bits to 1792 bits for 128 bit security due to the tight security reduction. While the original BLS signature length would be  $|\mathbb{G}_1| = 640$  bits, our signature is generated in terms of  $(|\mathbb{G}_1|, r)$ , where the generated signature length would be  $256 + 1 = 257$  bits as an additional bit is transmitted as  $r$ . Therefore, our signature length is 383 bits shorter compared to the original BLS signature at the same security level.

## 4.2 Tight Reduction as an Upgrade

The proposed BLS variant achieves the *seuf-cma* security besides having a tight reduction to the co-CDH problem. Although sharing the same security benefits as the Bellare et al.'s variant [6], ours is backward compatible while theirs are not. Therefore, theirs cannot be used to perform a “patching” to existing applications and standards [25] that are using the BLS signature.

For instance, our variant can be used on the aggregate BLS signatures as in [6]. Besides that, the variant can be also applied directly on Cha-Cheon's identity based signature (IBS) scheme [9] to further tighten the security of their scheme. Moreover, the variant is also applicable for works using the BLS signature for practical applications to have a tighter security, such as for the cloud storage [30] and public auditing [31].

## 4.3 Coron's BF-IBE

In [12], Coron proposed two variants of the BF-IBE based on the Decisional Square Bilinear Diffie-Hellman (D-Square-BDH) assumption and the Decisional

Bilinear Diffie-Hellman (DBDH) assumption. Both variants have a tight security with a loss of 1 bit due to the use of a random salt  $y \leftarrow \mathbb{Z}_q^*$  in the user secret key. Based on our proof in Sect. 3.3, it can be noticed that our method of using a 1 bit  $r$  can be applied on Coron's method of using the random salt  $y$  as well. By doing so, the length of their user secret key can be reduced without affecting the security tightness.

## 5 Conclusion

In this paper, we proposed a variant of the BLS signature scheme with a tight security reduction based on the co-CDH assumption. The new scheme has backward compatibility property and can be imposed directly on the original BLS signature scheme as an upgrade. Besides that, our scheme is upgraded to a stronger security notion compared to the original BLS scheme.

**Acknowledgment.** The authors would like to thank the Malaysia government's Fundamental Research Grant Scheme (FRGS/2/2014/ICT04/MMU/03/1) for supporting this work.

## References

1. Boneh, D., Boyen, X.: Short signatures without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) *EUROCRYPT 2004*. LNCS, vol. 3027, pp. 56–73. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-24676-3\\_4](https://doi.org/10.1007/978-3-540-24676-3_4)
2. Bao, F., Deng, R.H., Zhu, H.F.: Variations of Diffie-Hellman problem. In: Qing, S., Gollmann, D., Zhou, J. (eds.) *ICICS 2003*. LNCS, vol. 2836, pp. 301–312. Springer, Heidelberg (2003). [https://doi.org/10.1007/978-3-540-39927-8\\_28](https://doi.org/10.1007/978-3-540-39927-8_28)
3. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) *CRYPTO 2001*. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). [https://doi.org/10.1007/3-540-44647-8\\_13](https://doi.org/10.1007/3-540-44647-8_13)
4. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) *EUROCRYPT 2003*. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003). [https://doi.org/10.1007/3-540-39200-9\\_26](https://doi.org/10.1007/3-540-39200-9_26)
5. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. In: Boyd, C. (ed.) *ASIACRYPT 2001*. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001). [https://doi.org/10.1007/3-540-45682-1\\_30](https://doi.org/10.1007/3-540-45682-1_30)
6. Bellare, M., Namprempre, C., Neven, G.: Unrestricted aggregate signatures. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) *ICALP 2007*. LNCS, vol. 4596, pp. 411–422. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-73420-8\\_37](https://doi.org/10.1007/978-3-540-73420-8_37)
7. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of 1st ACM Conference on Computer and Communications Security – ACM CCS 1993, pp. 62–73. ACM (1993)
8. Bellare, M., Rogaway, P.: The exact security of digital signatures-how to sign with RSA and Rabin. In: Maurer, U. (ed.) *EUROCRYPT 1996*. LNCS, vol. 1070, pp. 399–416. Springer, Heidelberg (1996). [https://doi.org/10.1007/3-540-68339-9\\_34](https://doi.org/10.1007/3-540-68339-9_34)

9. Choon, J.C., Hee Cheon, J.: An identity-based signature from gap Diffie-Hellman groups. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 18–30. Springer, Heidelberg (2003). [https://doi.org/10.1007/3-540-36288-6\\_2](https://doi.org/10.1007/3-540-36288-6_2)
10. Chatterjee, S., Hankerson, D., Knapp, E., Menezes, A.: Comparing two pairing-based aggregate signature schemes. *Des. Codes Cryptogr.* **55**(2), 141–167 (2010). Springer
11. Coron, J.-S.: Optimal security proofs for PSS and other signature schemes. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 272–287. Springer, Heidelberg (2002). [https://doi.org/10.1007/3-540-46035-7\\_18](https://doi.org/10.1007/3-540-46035-7_18)
12. Coron, J.S.: A variant of Boneh-Franklin IBE with a tight reduction in the random oracle model. *Des. Codes Cryptogr.* **50**(1), 115–133 (2009)
13. Diffie, W., Hellman, M.: New directions in cryptography. *IEEE Trans. Inf. Theory* **22**, 644–654 (1976)
14. Kerry, C.F., Director, C.R.: FIPS PUB 186-4 Federal Information Processing Standards Publication Digital Signature Standard (DSS), FIPS Publication (2013)
15. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985). [https://doi.org/10.1007/3-540-39568-7\\_2](https://doi.org/10.1007/3-540-39568-7_2)
16. Goh, E.-J., Jarecki, S.: A signature scheme as secure as the Diffie-Hellman problem. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 401–415. Springer, Heidelberg (2003). [https://doi.org/10.1007/3-540-39200-9\\_25](https://doi.org/10.1007/3-540-39200-9_25)
17. Goldwasser, S., Micali, S.: Probabilistic encryption. *J. Comput. Syst. Sci.* **28**(2), 270–299 (1984). Springer, Heidelberg
18. Koblitz, N., Menezes, A.: Another look at “Provable Security”. II. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 148–175. Springer, Heidelberg (2006). [https://doi.org/10.1007/11941378\\_12](https://doi.org/10.1007/11941378_12)
19. Koblitz, N., Menezes, A.J.: The random oracle model: a twenty-year retrospective. *Des. Codes Cryptogr.* **77**(2–3), 587–610 (2015)
20. Kiltz, E., Masny, D., Pan, J.: Optimal security proofs for signatures from identification schemes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 33–61. Springer, Heidelberg (2016). [https://doi.org/10.1007/978-3-662-53008-5\\_2](https://doi.org/10.1007/978-3-662-53008-5_2)
21. Katz, J., Wang, N.: Efficiency improvements for signature schemes with tight security reductions. In: ACM – CCS 2003, pp. 155–164 (2003)
22. Lacharité, M.S.: Security of BLS and BGJS signatures in a multi-user setting. In: Advances in Cryptology 2016 – ARCTICCRYPT 2016, vol. 2, pp. 244–261. Springer, Heidelberg (2016)
23. Lu, S., Ostrovsky, R., Sahai, A., Shacham, H., Waters, B.: Sequential aggregate signatures and multisignatures without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 465–485. Springer, Heidelberg (2006). [https://doi.org/10.1007/11761679\\_28](https://doi.org/10.1007/11761679_28)
24. Liu, C., Ranjan, R., Zhang, X., Yang, C., Georgakopoulos, D., Chen, J.: Public auditing for big data storage in cloud computing-a survey. In: 2013 IEEE 16th International Conference on Computational Science and Engineering (CSE), pp. 1128–1135 (2013)
25. Moody, D., Peralta, R., Perlner, R., Regenscheid, A., Roginsky, A., Chen, L.: Report on pairing-based cryptography. *J. Res. Nat. Inst. Stand. Technol.* **120**, 11–27 (2015)
26. Barker, E., Barker, W., Burr, W., Polk, W., Smid, M.: Recommendation for key management-part 1: general (revised.) In: NIST Special Publication (2006)

27. Pereira, G.C., Simplício, M.A., Naehrig, M., Barreto, P.S.: A family of implementation-friendly BN elliptic curves. *J. Syst. Softw.* **84**(8), 1319–1326 (2011)
28. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **21**(2), 120–126 (1978). ACM
29. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, New York (1990). [https://doi.org/10.1007/0-387-34805-0\\_22](https://doi.org/10.1007/0-387-34805-0_22)
30. Wang, Q., Wang, C., Li, J., Ren, K., Lou, W.: Enabling public verifiability and data dynamics for storage security in cloud computing. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 355–370. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-04444-1\\_22](https://doi.org/10.1007/978-3-642-04444-1_22)
31. Zhang, F., Safavi-Naini, R., Susilo, W.: An efficient signature scheme from bilinear pairings and its applications. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 277–290. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-24632-9\\_20](https://doi.org/10.1007/978-3-540-24632-9_20)



# Quantum Authentication Scheme Based on Fingerprint-Encoded Graph States

Fei Li<sup>1</sup>, Ying Guo<sup>1(✉)</sup>, and Jiankun Hu<sup>2</sup>

<sup>1</sup> School of Information Science and Engineering, Central South University,  
Changsha 410083, China  
[yingguo@csu.edu.cn](mailto:yingguo@csu.edu.cn)

<sup>2</sup> School of Engineering and Information Technology,  
University of New South Wales at Australian Defence Force Academy,  
Canberra, ACT 2610, Australia

**Abstract.** We demonstrate an improved quantum authentication scheme which involves fingerprint recognition and quantum authentication. This scheme is designed to solve the practical problem in knowledge-based quantum authentication systems. It can satisfy the requirement of secure remote communication by using fingerprint-encoded graph states. The encoded graph states, which determine the preferred legitimate participants in the deterministic network, enable the facility of the implementable fingerprint-based authentication. The fingerprint template used for authentication in this scheme is of revocability and diversity. Security analysis shows that the proposed scheme can effectively defend various attacks including forgery attack, intercept-resend attack and man-in-the-middle attack. What's more, this novel scheme takes advantages of the merits in terms of both fingerprint recognition and quantum authentication, rendering it more secure, convenient and practical for users than its original counterpart, knowledge-based quantum authentication.

**Keywords:** Fingerprint · Graph state · Authentication · Security  
Quantum cryptography

## 1 Introduction

Most of the classical authentication algorithms depend on computational complexity and intractable mathematical problems [1,2]. However, with the rapid development of quantum technology, and especially the realization of a quantum computer, the classical algorithms may be broken, and thus the conventional authentication systems, including fingerprint recognition, will be in potential danger. Thus, a new authentication approach, namely the quantum authentication, comes into being. The main argument in favor of quantum authentication

---

Project supported by National Natural Science Foundation of China (Grant No. 61379153, 61572529).

originates from its tantalizing promise of providing unconditional security and detection against eavesdropping, due to the fundamental properties of quantum mechanics [3,4].

Recently, several quantum authentication protocols have been proposed in both theoretics and implementations. Dušek et al. [5] put forward an authentication protocol which combines quantum key distribution and classical identification procedure. Ljunggren et al. [6] proposed an authority-based user authentication system in quantum key distribution. Zhang et al. [7] presented an one-way quantum identity authentication protocol based on ping-pong technique and property of quantum controlled-NOT gate. Also, based on ping-pong technique, Yuan et al. [8] proposed an authentication protocol by using single-particle states. Chang et al. [9] presented an authentication protocol based on three-particle W state and quantum one-time pad. Naseri proposed a revisiting quantum authentication scheme based on entanglement swapping [10].

Nevertheless, the quantum authentication protocols which have been proposed are basically based on what you know. With the extensive application of quantum authentication and the deep development of informationization, authentication mechanisms based on what you know won't suffice to verify a person's identity [11]. Because, inevitably, these quantum authentication protocols will suffer the same trouble as those conventional and knowledge-based authentication protocols. For instance, with the development of the network, more and more people will need to remember a large of number of passwords, such as for online-banking, e-mail, social networks and so on, which is evidently inconvenient and makes users prone to errors. Thus, in order to memorize better, the authentication information tends to be short, which readily leads to security issues. However, if the authentication information is long for safety concerns, it will be easily forgotten. With practical application of quantum authentication, this problem has to be solve. Fortunately, combining fingerprint recognition and quantum authentication can ingeniously solve this problem, because only with the scanning of users finger over the sensor, the identity authentication process will be completed. However, the detailed realization of fingerprint-based quantum authentication protocols has not been discussed yet.

In the past decades, various types of fingerprint recognition methods have been proposed [12–15]. Comparing to other biometric traits, fingerprint has its own unique characteristics. There are no two identical fingerprints in the world so that other people can't pretend to be legitimate users. Moreover, fingerprint identifiers cannot be easily misplaced, forged or shared, which guarantees the security of fingerprint recognition. Thus, fingerprint-based quantum authentication not only possesses the advantage of unconditional security and detection against eavesdropping during the remote transmission, but also it is more convenient and practical than the knowledge-based authentication. The proposed fingerprint-based authentication methods can be divided into two categories, namely alignment-based [16] and alignment-free [17] approaches. For the alignment-based approach, a registration point(core) is required to align the

fingerprint image before further processing. In contrast to the alignment-based approach, no registration point is needed in the alignment-free approach.

In this paper, we propose a practical quantum authentication protocol using the fingerprint-encoded graph states. Graph state, as a special entangled quantum state, can be expressed by a mathematical graph whose vertices and edges are superb resources for establishing an elegant quantum network [19]. Compared to other quantum states, using the fingerprint-encoded graph states to transmit messages has several peculiar advantages. On one hand, graph states are the most easily available multipartite quantum states [20, 21]. On the other hand, each graph state can be represented with a mathematical graph so that it is conducive for us to understand how information spreads.

The rest of the paper is structured as follows. Section 2 details the framework of our authentication protocol. Section 3 shows the security of the proposed protocol. Finally, the conclusion is drawn in Sect. 4.

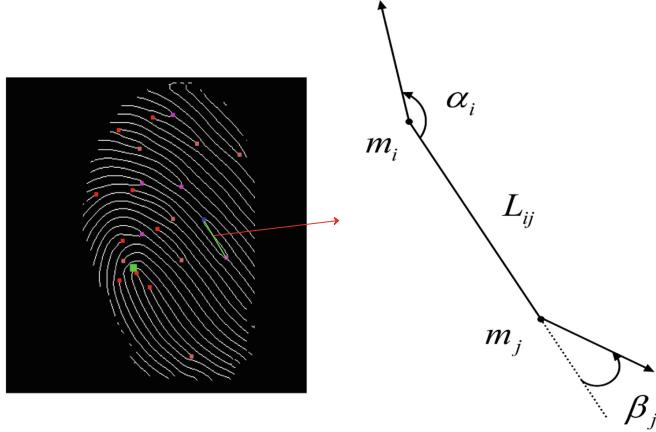
## 2 The Authentication Scheme with the Fingerprint-Encoded Graph States

### 2.1 Binary Representation Generation of Authentication

Fingerprint is an important feature of human beings. The word *fingerprint* is popularly perceived as synonymous with individuality. The most significant structural characteristic of a fingerprint is the pattern of interleaved ridges and valleys. Usually, ridges run smoothly in parallel but exhibit one or more regions where they assume distinctive shapes (characterized by high curvature, frequent ridge terminations, etc.). These regions, called singularities or singular regions. According to the characteristic of singular regions, fingerprint can be classified into five categories: left loop, right loop, whorl, arch and tented arch. The core point corresponds to the center of the north most loop or whorl type singularity. For arch fingerprints and tented-arch fingerprints, the core point is difficult to be defined. Even, for loop fingerprints and whorl fingerprints, sometimes it is also difficult to correctly locate the core point due to the high variability of fingerprint patterns during capture. Thus, the type of the selected fingerprint has a great influence on the security of alignment-based authentication, because the accuracy of alignment-based authentication depends highly on the core point, while the security of alignment-free authentication is independent of the type of the selected fingerprint.

In what follows, we describe an alignment-free revocable fingerprint template generation. The fingerprint can be represented by a set of minutiae points extracted from the fingerprint image, which is denoted by  $m_i = \{x_i, y_i, \theta_i\}$ , where  $x_i$ ,  $y_i$  and  $\theta_i$  are the  $x$ ,  $y$  coordinates and the orientation of the  $i^{th}$  minutiae, respectively. Due to various factors during fingerprint capture, single minutiae point is readily subjected to elastic deformation, while a minutiae pair which is formed by two minutiae points tends to be immune to nonlinear distortion. The procedure of extracting binary information from minutiae pairs is listed below.

**Step 1. Features extracted from minutiae pairs.** We connect a pair of minutiae by a straight line. The invariant features used in our work are the length  $L$  between the two minutiae points and two angles, denoted by  $\alpha$  and  $\beta$ , between the orientation of each minutiae and the straight line. Let  $F_{ij}$  represent the invariant feature extracted from a minutiae pair which is made up of the minutiae  $m_i$  and  $m_j$ . As shown in Fig. 1,  $F_{ij} = \{L_{ij}, \alpha_i, \beta_j\}$ .



**Fig. 1.** The invariant feature extracted from the minutiae pair  $(m_i, m_j)$ .  $L_{ij}$  is the length between the two minutiae pair.  $\alpha_i$  and  $\beta_j$  are the angles between the straight line and the orientations of the minutiae  $m_i$  and  $m_j$ , respectively.

In order to obtain the value of  $F_{ij}$ , the values  $X_{ij}$  and  $Y_{ij}$  need to be calculated as follows:

$$\begin{bmatrix} X_{ij} \\ Y_{ij} \end{bmatrix} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \begin{bmatrix} x_j - x_i \\ -(y_j - y_i) \end{bmatrix}. \quad (1)$$

Therefore, we have

$$L_{ij} = \sqrt{X_{ij}^2 + Y_{ij}^2}, \quad \alpha_i = \arctan(\frac{Y_{ij}}{X_{ij}}), \quad \beta_j = \alpha_i + \theta_j - \theta_i. \quad (2)$$

**Step 2. Quantization of the invariant features.** In order to resist the non-linear distortion brought during the image acquisition, the features need to be quantized. An appropriate quantization step, that is the number of bits to quantize each feature, should be judiciously determined by experiments. Because, the accuracy of the system is closely dependent on quantization. Let  $len$ ,  $a_1$  and  $a_2$  represent the length of binary representation of  $L_{ij}$ ,  $\alpha_i$  and  $\beta_j$ . So each pair of minutiae can be represented by a bit string whose length is  $N = len + a_1 + a_2$ .

**Step 3. Generation of the bit-string fingerprint representation.** After quantizing all the minutiae pairs, we convert the binary representation into the

decimal form to be the index of the histogram and then calculate the histogram of minutiae pairs. At the beginning, the histogram is made up of  $2^N$  zeros. Then, we inspect each index and the value in the histogram corresponding to the index is added by one. Finally, we binarize the histogram with a simple rule that the value 1 in the histogram is retained whereas the rest of values is set to 0. So we get the bit-string representation for the fingerprint.

**Step 4. Permutation of the binary string.** The binary string generated in Step 3 is vulnerable and may be employed to access another fingerprint-based system. Thus, to protect the privacy of users, the string needs to be permuted. The permutation is based on the unique key, which is assigned to each user. In other words, different users employ different manners during permutation. The key for permutation is random so that the permuted template can't reveal any information about the original template without the user-specific key.

## 2.2 Information Transmission Using Graph States

The graph state is an entangled state, which can be described with a simple undirected graph mathematically [22]. An undirected graph  $G = \{V, E\}$  is made up of a set of  $n$  vertices and a set of edges  $E = \{e_{ij} = (v_i, v_j)\}$ , where  $v_i$  and  $v_j$  are neighbors when there is a edge connecting them. In a graph state, each vertex represents a qubit. All of graph states generate from an initial state [23]

$$|+\rangle^{\otimes n} = H^{\otimes n}|0\rangle^{\otimes n}, \quad H = |+\rangle\langle 0| + |- \rangle\langle 1|, \quad (3)$$

where  $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$  and  $H$  is the Hadamard operator. After applying the two-qubit controlled-phase gate (denoted by  $CZ$ ) on all pairs of qubits whose corresponding vertices are adjoining, we can get an initial graph state

$$|G\rangle = \prod_{(v_i, v_j) \in E} CZ_{(v_i, v_j)} |+\rangle^{\otimes n}, \quad (4)$$

where

$$CZ|kk'\rangle = (-1)^{kk'}|kk'\rangle, \quad k, k' \in \{0, 1\}, \quad |kk'\rangle \in H_2^{\otimes 2}. \quad (5)$$

The order of applying  $CZ$  gates is unimportant, because the operation possesses the exchange property that establishes the deterministic quantum network.

Usually, each vertex of a graph state is labeled by an index. For instance, the index of the vertex  $v_i$  is  $i$ . In order to make better use of graph states, we label each vertex with two more bits. So, the vertex  $v_i$  is labeled as  $(i, l_{i1}, l_{i2})$ . The two extra label bits are employed to transmit classical information. For ease of understanding, we define the quantities  $l_{i*} = (l_{i1}, l_{i2})$  for the  $i^{th}$  vertex,  $l_{*j} = (l_{1j}, l_{2j}, \dots, l_{nj})$  for the  $j^{th}$  bit of all vertices, and  $l = (l_{11}, l_{12}, l_{21}, l_{22}, \dots, l_{n1}, l_{n2})$  for the graph state. By the way, when the vertex has no label,  $l_{i*}$  is set to  $(0, 0)$ . With these additional labels, we can obtain the labeled graph state by

$$|G_l\rangle = \bigotimes_i (X_i^{l_{i1}} Z_i^{l_{i2}}) |\tilde{G}\rangle, \quad |\tilde{G}\rangle = \bigotimes_{j|v_j \in V} S_j |G\rangle, \quad (6)$$

where  $X = |0\rangle\langle 1| + |1\rangle\langle 0|$ ,  $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$  and  $S = |0\rangle\langle 0| - i|1\rangle\langle 1|$ . The partial phase shift gate [19] is employed to prevent the system from eavesdropping. For the sake of encoding and simplifying the manipulation ,we only retain local  $Z$  gates and introduce another kind of graph state, called the encoded graph state

$$|G_{l_{*2}}\rangle = \bigotimes_i Z_i^{l_{i2}} |\tilde{G}\rangle. \quad (7)$$

When  $l_{i1}$  is equal to 0 for  $\forall i \in V$  in a labeled graph state, the labeled graph state becomes a encoded graph state, which can be expressed in the stabilizer formalism [22]:

$$K_i |G_{l_{*2}}\rangle = (-1)^{l_{i2}} |G_{l_{*2}}\rangle, \quad K_i = X_i \bigotimes_{(v_i, v_j) \in E} Z_j. \quad (8)$$

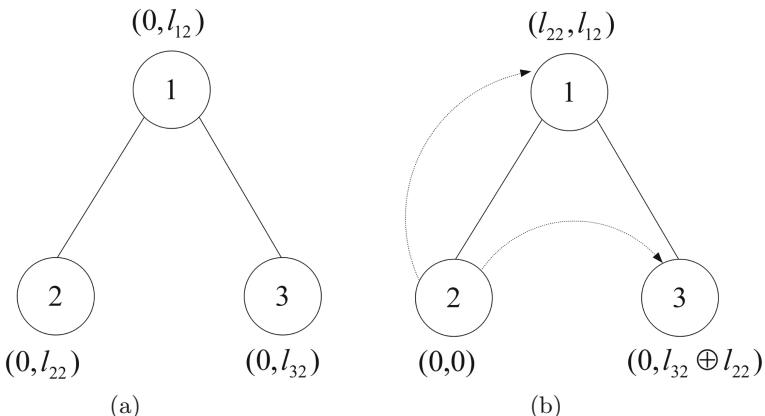
In the proposed protocol, the three-qubit labeled graph state, as depicted in Fig. 2, is used for the authentication information processing, which can be given by

$$|G_{l_{*2}}\rangle = \bigotimes_{i=1}^3 Z_i^{l_{i2}} |\tilde{G}\rangle, \quad |\tilde{G}\rangle = \frac{1}{\sqrt{2}}(|0+\rangle + |1-\rangle). \quad (9)$$

Consequently, the encoded graph state can be described by stabilizers

$$K_1 = X_1 \bigotimes Z_2 \bigotimes Z_3, \quad K_2 = Z_1 \bigotimes X_2 \bigotimes I_3, \quad K_3 = Z_1 \bigotimes I_2 \bigotimes X_3, \quad (10)$$

with eigenvalues  $(l_{12}, l_{22}, l_{32})$ .



**Fig. 2.** A labeled graph state for three players and information transmission using the stabilizer. (a) The initial labeled graph state with eigenvalues  $(l_{12}, l_{22}, l_{32})$ . (b) By performing the stabilizer  $K_1$  on the encoded graph state, the bit  $l_{22}$  is transmitted to other parties, i.e.,  $l_{1*} = (l_{22}, l_{12})$ ,  $l_{2*} = (0, 0)$ ,  $l_{3*} = (0, l_{32} \oplus l_{22})$ .

By performing above-derived stabilizers on the graph state, we can accomplish the encoded process. For example, acting upon the encoded graph state by the first stabilizer  $K_1$  results in

$$\begin{aligned}
K_1^{l_{22}} |G_{l_{*2}}\rangle &= (X_1^{l_{22}} \bigotimes Z_2^{l_{22}} \bigotimes Z_3^{l_{22}}) (\bigotimes_{i=1}^3 Z_i^{l_{i2}} |\tilde{G}\rangle) \\
&= X_1^{l_{22}} Z_1^{l_{12}} \bigotimes I_2 \bigotimes Z_3^{l_{22}} Z_3^{l_{32}} |\tilde{G}\rangle \\
&= |G_{l=(l_{22}, l_{12}, 0, 0, 0, l_{32} \oplus l_{22})}\rangle \\
&= (-1)^{l_{12}l_{22}} |G_{l_{*2}}\rangle.
\end{aligned} \tag{11}$$

It implies that the bit  $l_{22}$  is transmitted from the vertex  $v_2$  to  $v_1$  and  $v_3$ . After that, we take local Pauli measurements in bases  $\{X_i, Y_i, Z_i\}$  to get one-bit outcomes  $s_i^X$ ,  $s_i^Y$  and  $s_i^Z$  [19, 24], respectively. When the measurement outcome is the value 1,  $s_i^\alpha$  is set to the value 0. Otherwise,  $s_i^\alpha$  is assigned the value 1. Finally, we access the labeled bits by using the relations

$$l_{22} = s_1^Z \bigoplus s_2^X, \quad l_{32} = s_1^Z \bigoplus s_3^X. \tag{12}$$

### 2.3 Implementation of Authentication Processing

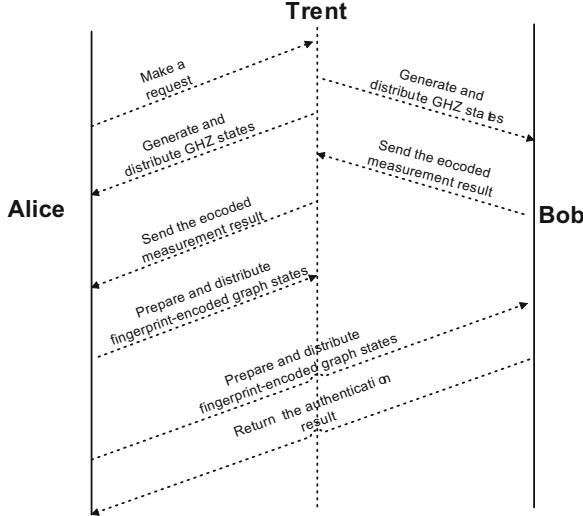
Suppose that Alice is the user, Bob is the server of a certain application which possesses the users' fingerprint templates, and Trent acts as the reliable third party. Then we detail the practical implementation of the fingerprint-based quantum authentication in the deterministic network.

**Enrollment Phase.** In this phase, the system needs to generate a fingerprint template for Alice. Firstly, Alice's fingerprint characteristic is sensed and captured by a fingerprint scanner to produce a fingerprint sample. Usually, a quality checking operation is first implemented to guarantee that the acquired sample is reliable enough for successive processing. Then, we extract the minutiae from the fingerprint and generate the binary representation (denoted by  $E^n(x)$ ). Taking  $E^n(x)$  as the control parameter, Alice's fingerprint template can be constructed as

$$|f_x\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^n (-1)^{E_i(x)} |i\rangle, \tag{13}$$

which is kept by Bob. After that, Alice and Bob obtain their keys  $K_a$  and  $K_b$  through quantum key distribution in quantum networks, where  $K_a$  is the key shared between Alice and Trent, and  $K_b$  is the key between Bob and Trent.

**Authentication Phase.** In this phase, Alice submits a request to Trent and claims that she is Alice and wants to communicate with Bob. The authentication procedure among Alice, Bob and Trent, as shown in Fig. 3, is listed below.



**Fig. 3.** The procedure of authentication. After receiving Alice's request, Trent prepares GHZ states and distributes them to Alice and Bob, respectively. Then, there is a qualification examination among Alice, Bob and Trent. After that, Alice inputs her fingerprint and the system generates the authentication information, which is encoded into graph states and sent to Bob and Trent by performing stabilizers on graph states. Finally, Bob compares the authentication information with the enrollment template and returns the authentication result.

*Step 1.* Trent generates  $k$  GHZ tripartite states after receiving Alice's request. Two particles of each GHZ tripartite state are transmitted to Alice and Bob respectively and the remaining one is kept by himself. Bob and Trent randomly measure their own particles, leading to Bob's measurement result  $R_b$  and Trent's measurement result  $R_t$ .

*Step 2.* Bob encrypts  $R_b$  with the key  $K_b$

$$y_b = K_b(R_b), \quad (14)$$

and then sends  $y_b$  and  $R_b$  to Trent through a classical channel.

*Step 3.* Trent decrypts  $y_b$  with the key  $k_b$

$$R'_b = K_b(y_b). \quad (15)$$

If  $R'_b$  is equal to  $R_b$ , it implies that Bob is honest. Otherwise, the protocol is aborted. After examining the qualification of Bob, Trent encrypts  $R_b$  and  $R_t$  with the key  $K_a$

$$y_t = K_a(R_b, R_t), \quad (16)$$

and then sends  $y_t$  to Alice through a classical channel.

*Step 4.* Alice decrypts  $y_t$  and selects corresponding measurement bases to measure her particles according to  $R_b$  and  $R_t$ . After that, she compares her measurement result  $R_a$  with  $R_b$  and  $R_t$ . If the yielded results satisfy the correlation of the GHZ tripartite states, Alice executes the following authentication procedures. Otherwise, Alice supposes that Trent is dishonest or the channel is insecure and then the protocol will be aborted.

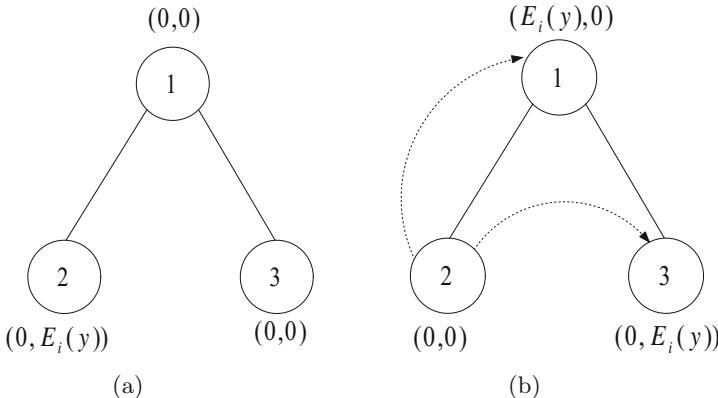
*Step 5.* Alice inputs her fingerprint and the system generates a binary representation (denoted by  $E^n(y)$ ) for the fingerprint with the afore-described method in enrollment phase.

*Step 6.* Alice prepares  $n$  three-qubit encoded graph states. The label of the  $i^{th}$  encoded graph state is  $l^i = (0, 0, 0, E_i(y), 0, 0)$ . Alice retains the second particle of each graph state and distributes the first particle of each graph state to Trent. And the other particles are sent to Bob.

As shown in Fig. 4, applying the first stabilizer  $K_1^i$  on the  $i^{th}$  encoded graph state for all  $i$  yields

$$(K_1^i)^{E_i(y)} |G_{l^i=(0,0,0,E_i(y),0,0)}\rangle = |G_{l^i=(E_i(y),0,0,0,0,E_i(y))}\rangle, \quad (17)$$

where  $1 \leq i \leq n$ . It indicates that the bit  $E_i(y)$  is transferred from Alice to Bob and Trent.



**Fig. 4.** Transmission of the representation information. (a) The initial encoded graph state with the label  $l = (0, 0, 0, E_i(y), 0, 0)$ . (b) Equivalent state with the representation bit  $E_i(y)$  transmitted from Alice to Bob and Trent.

*Step 7.* Bob applies measurement operations on the yielded encoded graph state  $|G_{l^i=(E_i(y),0,0,0,0,E_i(y))}\rangle$ , and obtains the binary representation  $E^n(y)$  of Alice's fingerprint. According to  $E^n(y)$ , the authentication qubit (denoted by  $|f_y\rangle$ ) is generated by

$$|f_y\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^n (-1)^{E_i(y)} |i\rangle. \quad (18)$$

*Step 8.* Bob compares the authentication qubit with the enrollment qubit. The similarity score will be obtained by calculating their inner product

$$Score = (\lvert f_x \rangle, \lvert f_y \rangle). \quad (19)$$

If the score is above the threshold, Bob regards Alice as a legitimate user. Otherwise, Bob rejects Alice's request.

In the above-mentioned authentication scheme, the encoding operations of the stabilizers is simple and readily implemented, and thus the transmission and measurement of the authentication information can be achieved handily by performing local unitary operations rather than the complicated joint operations in the traditional schemes.

### 3 Security Analysis

#### 3.1 Forgery Attack

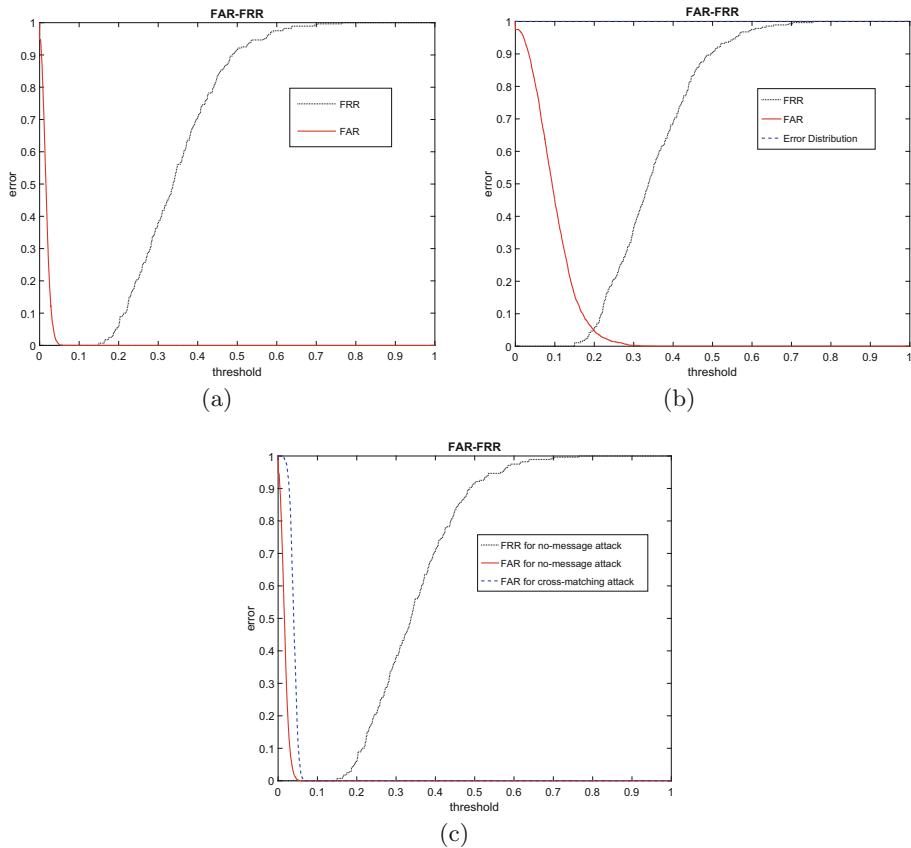
The forgery attack is focused on the strategy of non-message attack, lost-key attack and cross-matching attack. The fingerprint database FVC2002 DB1 is used to test our proposed scheme. In order to evaluate the accuracy of a fingerprint-encoded authentication system, the false acceptance rate (FAR) and the false rejection rate (FRR) should be introduced. FAR is the probability of mistaking two fingerprints from two different fingers to be from the same finger, whereas FRR is the probability of mistaking two fingerprints from the same finger to be from two different fingers.

*Non-message attack:* The attacker, Eve, pretends to be a legitimate user and attempts to pass through the authentication without any valid information. In such a situation, the probability of a successful attack is  $P = (\frac{1}{2})^N$ . Obviously, when N is big enough, the probability is  $P = (\frac{1}{2})^N \approx 0$ . To demonstrate the performance of the proposed protocol under non-message forgery attack, every user is assigned with a unique user-specific key in our experiment. As shown in Fig. 5(a), if we select an appropriate threshold, the ideal result, where FAR and FRR are both equal to 0, can be obtained. It indicates that the probability of a successful attack is 0 in practical application.

*Lost key attack:* It is the worst case where the user's key is known by Eve. In conventional knowledge-based quantum authentication, Eve can successfully pretend herself to be Alice to communicate with Bob when the user's key is compromised. In other words, the probability of a successful lost key attack is 100% in this case. However, in our proposed scheme, even if the user's key has lost, because the key kept by the user is random and independent of the user's fingerprint, Eve still requires a large number of attempts to uncover the binary fingerprint representation. To demonstrate the performance under lost key attack, we performed this experiment by assigning the same key to all users. As shown in Fig. 5(b), it demonstrates that even if the user's key is stolen by Eve, the probability of mistaking the adversary as a legitimate

user is still quite low in our proposed scheme. However, once the user's key has lost in conventional knowledge-based authentication, the system will be completely exposed to the adversary.

*Cross-matching attack:* Eve attempts to employ the template generated in one application to have access to other applications where the template owner has registered. Because the key for permutation is random, two templates generated from the same user won't match. This case was simulated by using different keys to permute the binary string generated from the same finger-print impression and then we calculated their similarities. Figure 5(c) shows that the FAR curve in this experiment is similar to the FAR curve for non-message attack and there is a clear separation between the FAR curve and



**Fig. 5.** The probability of a successful forgery attack. Experimental parameters is set as:  $len = 6$ ,  $a_1 = 5$ , and  $a_2 = 5$ . (a) The FAR-FRR distribution for non-message attack. (b) The FAR-FRR distribution for lost key attack. The dash blue curve gives the error distribution for the conventional knowledge-based quantum authentication, namely the false acceptance rate under lost key attack. (c) The FAR-FRR distribution for cross-matching attack. (Color figure online)

the FRR curve, as if the templates originating from the same user in different applications are generated from different users. Thus, the cross-matching attack cannot be achieved. Meanwhile, it means revocability and diversity that when an enrolled template is compromised, a new fingerprint template can be regenerated, and it cannot match with the compromised template even though both are generated from the same fingerprint. Due to the revocability and the diversity of the template, the drawback in fingerprint recognition, that the number of each people's fingerprints which are used for authentication is small and limited, can be solved quite well. Furthermore, the original biological authentication information can be well protected from eavesdropping during remote transmission. What's more, compared to the conventional knowledge-based authentication, the users can get rid of remembering a large number of passwords, which can't compromise security.

The user can regularly update the key which is used for permutation to generate a new template. As mentioned above, the new template won't match the old template so that the security of the fingerprint template can be enhanced and guaranteed.

### 3.2 Intercept-Resend Attack

In order to pass through the authentication, Eve may intercept the particles which Alice sends to Bob or Trent and then resend a forged sequence to Bob or Trent according to her measurement result. However, the label of each particle sent to Bob and Trent is  $(0, 0)$ , namely having no information about the binary fingerprint representation encoded into these particles. Thus, even if Eve measures the particles intercepted, she obtains nothing, namely

$$I(E, T) = 0, I(E, B) = 0. \quad (20)$$

where  $I(E, T)$  is the mutual information between Eve and Trent, and  $I(E, B)$  is the mutual information between Eve and Bob. What's more, the correlation between the particles intercepted and the particles kept by Alice is released, which affects transferring the label bits. Thus, even if Eve intercepts the particles, she cannot get any valid information and the disturbed actions can be detected by in the authentication phase.

### 3.3 Man-in-the-Middle Attack

To obtain the information which Alice sends to Bob, Eve disguises herself as Bob to communicate with Alice, and also plays the role of Alice to communicate with Bob. As we know, in quantum cryptography, there is a fundamental assumption that Eve cannot simultaneously obtain information on quantum channels and classical channels. Therefore, when Eve receives the particles which Trent sends to Alice or Bob and disguises herself to communicate with the other one, according to the assumption, she cannot obtain the information on the classical channels.

Furthermore we consider a situation that Eve plays the role of Trent. Because Eve don't know the correct key corresponding to Alice, the measurement results  $R_a$ ,  $R_b$  and  $R_t$  will not satisfy the correlation of the GHZ tripartite states. Therefore, Alice can find that Trent is dishonest and then the protocol will be aborted.

## 4 Conclusion

We have demonstrated an improved quantum authentication scheme based on fingerprint-encoded graph states. It has the advantages of both fingerprint recognition and quantum authentication in the remote deterministic quantum networks, which is more convenient, practical and secure than knowledge-based quantum authentication. There are two phases, namely enrollment phase and authentication phase, involved in our proposed scheme. In enrollment phase, the system generates the user's fingerprint template which is revocable and diverse, whereas in authentication phase the system generates the binary fingerprint representation for the user and then the binary information is transmitted using the fingerprint-encoded graph states. Security analysis shows that the proposed scheme can effectively defend various attacks including forgery attack, intercept-resend attack and man-in-the-middle attack.

## References

1. Niu, P., Chen, Y., Li, C.: Quantum authentication scheme based on entanglement swapping. *Int. J. Theor. Phys.* **55**, 1–11 (2016)
2. Jin, Z., Teoh, A.B.J., Ong, T., Tee, C.: A revocable fingerprint template for security and privacy preserving. *KISS Trans. Internet Inf. Syst.* **4**, 1327–1342 (2010)
3. Liu, B., Gao, F., Huang, W., Wen, Q.Y.: QKD-based quantum private query without a failure probability. *Sci. China Phys. Mech. Astron.* **58**, 100301 (2015)
4. Gaudiano, M., Osenda, O.: Entanglement in a spin ring with anisotropic interactions. *Int. J. Quant. Inf.* **13**, 1550057 (2015)
5. Dušek, M., Haderka, O., Hendrych, M., Myška, R.: Quantum identification system. *Phys. Rev. A* **60**, 149–156 (1998)
6. Ljunggren, D., Bourennane, M., Karlsson, A.: Authority-based user authentication and quantum key distribution. *Phys. Rev. A* **62**, 299–302 (2000)
7. Zhang, Z.S., Zeng, G.H., Zhou, N.R., Xiong, J.: Quantum identity authentication based on Ping-pong technique for photons. *Phys. Lett. A* **356**, 199–205 (2006)
8. Yuan, H., Liu, Y., Pan, G., Zhang, G., Zhou, J., Zhang, Z.: Quantum identity authentication based on Ping-pong technique without entanglements. *Quant. Inf. Process.* **13**, 2535–2549 (2014)
9. Chang, Y., Zhang, S., Yan, L., Li, J.: Deterministic secure quantum communication and authentication protocol based on three-particle W state and quantum one-time pad. *Sci. Bull.* **59**, 2835–2840 (2014)
10. Naseri, M.: Revisiting quantum authentication scheme based on entanglement swapping. *Int. J. Theoret. Phys.* **55**, 2428–2435 (2016)
11. Maltoni, D., Maio, D., Jain, A., Prabhakar, K.S.: *Handbook of Fingerprint Recognition*. Springer, London (2009). <https://doi.org/10.1007/978-1-84882-254-2>

12. Wang, Y., Hu, J.: Global ridge orientation modeling for partial fingerprint identification. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 72 (2011)
13. Jin, Z., Teoh, A.B.J., Ong, T.S., Tee, C.: Generating revocable fingerprint template using minutiae pair representation. In: 2nd International Conference on Education Technology and Computer, pp. 22–24. IEEE Press, New York (2010)
14. Yang, W., Hu, J., Wang, S.: A Delaunay quadrangle-based fingerprint authentication system with template protection using topology code for local registration and security enhancement. *IEEE Trans. Inf. Forensics Secur.* **9**, 1179–1192 (2014)
15. Wong, W.J., Teoh, A.B., Kho, Y.H., Wong, M.L.D.: Kernel PCA enabled bit-string representation for minutiae-based cancellable fingerprint template. *Pattern Recogn.* **51**, 197–208 (2016)
16. Ratha, N.K., Chikkerur, S., Connell, J.H., Bolle, R.M.: Generating cancelable fingerprint templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**, 561–572 (2007)
17. Lee, C.H., Choi, C.Y., Toh, K.A.: Alignment-free cancelable fingerprint templates based on local minutiae information. *IEEE Trans. Syst. Man Cybern.* **37**, 980–992 (2007)
18. Thomas, A.O., Ratha, N.K., Connell, J.H., Bolle, R.M.: Comparative analysis of registration based and registration free methods for cancelable fingerprint biometrics. In: 19th International Conference on Pattern Recognition, pp. 8–11. IEEE Press, New York (2008)
19. Markham, D., Sanders, B.C.: Graph states for quantum secret sharing. *Phys. Rev. A* **78**, 042309 (2011)
20. Lu, C.Y., Zhou, X.Q., Gühne, O., Gao, W.B., Zhang, J., Yuan, Z.S., Goebe, A., Yang, T., Pan, J.: Experimental entanglement of six photons in graph states. *Nat. Phys.* **3**, 91–95 (2007)
21. Walther, P., Resch, K.J., Rudolph, T., Schenck, E., Weinfurter, H., Vedral, V., Aspelmeyer, M., Zeilinger, A.: Experimental one-way quantum computing. *Nature* **434**, 169 (2005)
22. Nest, M.V.D., Dehaene, J., Moor, B.D.: An efficient algorithm to recognize local Clifford equivalence of graph states. *Phys. Rev. A* **70**, 423–433 (2004)
23. Hein, M., Eisert, J., Briegel, H.J.: Multi-party entanglement in graph states. *Phys. Rev. A* **69**, 666–670 (2003)
24. Keet, A., Fortescue, B., Markham, D., Sanders, B.C.: Quantum secret sharing with qudit graph states. *Phys. Rev. A* **82**, 4229–4231 (2010)



# Cooperative Information Security/ Cybersecurity Curriculum Development

Abdelaziz Bouras<sup>1()</sup>, Houssem Gasmi<sup>1,2</sup>, and Fadi Ghemri<sup>1</sup>

<sup>1</sup> Computer Science and Engineering Department, Qatar University,  
PO Box 2713 Doha, Qatar

{abdelaziz.bouras, houssem.gasmi, fadi.ghemri}@qu.edu.qa

<sup>2</sup> Disp Lab, Université Lumière Lyon 2, 160, Bd de L'université,  
69676 Lyon, Bron, France

**Abstract.** It is often difficult to meaningfully convey concepts like security incident management cycle, information sharing, cooperation, as well as the roles of people, processes and technology in information and cybersecurity courses. Such complexity requires immersive and interactive learning based on continuous cooperation between industry and academia. In this paper we highlight the ongoing industry/university cooperative effort towards an cooperative schema to enforce the Information Security and Cybersecurity Curriculum development within an existing Master of Computing.

**Keywords:** Cooperative education · Cybersecurity · Competency frameworks  
Mentorship · Ethical Hacking · Tabletop

## 1 Introduction

The past decade in Qatar has seen a rapid economic development with increasingly fast advances in many fields, particularly in new technologies, due to its fast growing economy. Qatar is currently building large infrastructures in various domains (services, industry, health, tourism, etc.) as part of the Qatar National Vision 2030 [1]. As the country has become more connected, traditional threats from hackers have been rapidly and continuously changing to include more malicious players such as terrorists, organized criminal networks, and in some cases foreign industrial/government cyber espionage. Critical infrastructures such as power plants, air-traffic and fuel refineries are increasingly targeted by cyber-attacks (ex. Shamoon virus crippled thousands of computers at Qatar's RasGas in 2012). The last must example is when Qatar's state news agency was hacked by anonymous party, this incident provoked a serious diplomatic crisis in the region.

From the education side, it is often difficult to meaningfully convey concepts like security incident management cycle, information sharing, cooperation, as well as the roles of people, processes and technology in information and cybersecurity courses. Such complexity requires continuous interaction between industry and academia to provide immersive and interactive learning experience to students. This will help building innovative solutions and developing new skills and competencies [2]. Qatar's academic community has an interesting young history of responding to the needs of

industry. Since few years the cybersecurity topics were highlighted by the Qatar National Research Fund (QNRF)<sup>1</sup> research programs, and dozens of programs have been funded. Some of these programs involve industry partners and stakeholders.

Globally, universities play an important role in multiple ways to contribute in the development and expansion of local competencies, especially in cyber security which is shown many lacks related to the content of curriculum, local universities are called to respond to the local needs, by providing efficient local competencies: through the provision of skilled graduates who become key players in local industry; through the conduct of long-term fundamental research that contributes to the science base and understanding available to private firms; through the promotion of an atmosphere of intellectual diversity that tolerates different approaches to the solution of technical problems; through direct collaboration with industry both on specific projects and in longer term relationships. Consequently, the state of Qatar has to invest in the security education, convinced by the idea of a very competitive future world, based on the capacity of countries to develop a well protected knowledge-based economy.

## 2 Need for Cybersecurity Competency Frameworks

Several paradigms related to information security and Cybersecurity dealing with the complex, multidisciplinary nature of the field have been defined. [3] for instance observed that cybersecurity comprises three planes of study:

- *Operations*: The day-to-day functioning of the information security task.
- *Governance*: The management of the cybersecurity function, including internal policies and procedures as well as law and policy.
- *Education/training*: Transfer of knowledge to cybersecurity professionals and users, ranging from teaching specific skills and competencies to providing systemic understanding and life-long learning.

Within a recent QNRF research project (PROSKIMA NPRP7-1883-5-289) [4], interviews have been conducted in Qatar with Information Technology (IT) representatives and security experts from major companies, organizations and ministries to have their feedback regarding the last point (Education/Training). These interviews highlighted the growing needs for enhanced competencies for Cybersecurity. Consequently, higher educational institutions need to reflect such evolution in their training curricula. Trained students will be able to tackle real world challenges efficiently.

Moreover, to have a clear picture and substantial update, a series of workshops titled “Industry Integrated Engineering and Computing Education Curriculum”<sup>2</sup> has been organized at Qatar University, through its Computer Science and Engineering (CSE) department, where professionals and experts from different organizations and industries met with the academics to identify relevant Cybersecurity competency frameworks, that need to be

---

<sup>1</sup> <https://www.qnrf.org/en-us/>.

<sup>2</sup> <http://www.qu.edu.qa/newsroom/Engineering/2nd-Industry-Integrated-Computing-Education-Curricula-Workshop-held>.

used to enhance existing university curricula or to introduce new integrative programs. Discussions and outcomes of this session are summarized in the Table 1 here below.

**Table 1.** Outcome from industry/academia brainstorming sessions

Activity	Competencies
Identify	<ul style="list-style-type: none"> <li>• Security Analysis(Industrial, cloud, IOT, Big Data)</li> <li>• Risk (Identify, User baseline, Perform risk assessment externally, Management, Governance)</li> <li>• Fault forecasting (Before, Lifetime)</li> </ul>
Protect	<ul style="list-style-type: none"> <li>• Design and Implement security mechanism (Authentication, Authorization, Confidentiality, Integrity)</li> <li>• Perform Security Auditing (Gaps, Vulnerabilities)</li> <li>• Follow levels of Compliance (Standard, Maturity, Training, Audit, Review)</li> </ul>
Detect	<ul style="list-style-type: none"> <li>• Deploy and configure intrusion detection system (IDS)/ Monitoring</li> <li>• Analyze Data (logs, application...) to detect threats</li> <li>• Monitor and detect incidents and threats</li> </ul>
Respond	<ul style="list-style-type: none"> <li>• Mitigate and stop attacks</li> <li>• Build a Proper Communication with stakeholders/ Media</li> <li>• Build Information Security incident response skills and awareness</li> </ul>
Recover	<ul style="list-style-type: none"> <li>• Perform all business continuity actions (fault Removal, Data Recovery)</li> </ul>
Cross- Cutting (Applies to all function above)	<ul style="list-style-type: none"> <li>• Understand the business process</li> <li>• Build Policies/Framework/standard/Compliance/ Processes</li> <li>• Be up to date regarding Technology (Mobile, Cloud, Cryptography, Electronics, Network, OS, PCs, Directory, Database, Analytics, Linux, Open Standard)</li> </ul>

### 3 Achievement Through Industry/University Cooperative Schema

It is commonly agreed that cybersecurity is more about processes than technology and the answer to the cyber-related security challenges are not solely about information technology and technical solutions but must also involve other related topics such as sociology, national defense, economics, political science, diplomacy, history, and many other social sciences [5].

In our case, and based on the competency needs highlighted earlier, we propose to enforce the current Master of Computing<sup>3</sup> with such vision. Because a global change in the structure of the curriculum needs long time due to several committees' validations (department, college, university), we believe that the achievement of such complete cooperative schema needs the adoption of at least two steps:

<sup>3</sup> Study plan: <http://www.qu.edu.qa/engineering/computer/programs/phd/studyPlan.php>

Courses: <http://www.qu.edu.qa/engineering/computer/programs/phd/cs/csListOfCourses.php>.

- Step1: Enforce the current curricula, by adding new cooperative tools (Mentorship, Reflexive Sessions, Industry Projects, Specific courses...). This is highlighted in this paper.
- Step2: Transform the Master's curricula to achieve collaborative sandwich programs (alternate stays of students between industry and university, for ex. Rotation every 2 weeks). A deep involvement of the stakeholders is necessary to shape such relationship and analyze additional challenges related to the certifications, etc. This step is beyond the scope of this paper.

In this following sections we mainly focus on the main components of the first step.

### 3.1 Project Mentorship

The aim of this component is to provide the students (trainees) with practical experience under the mentorship of two mentors, one from the university and the other from the company. While certainly useful, this approach has scalability issues, since finding mentors for all students is often difficult to achieve. In our case, this should not be a problem to be initiated for the Master of Computing due to its reasonable size. However, its generalization to the Bachelor sections will definitely meet such limitations. Solutions could be found within the TIEE (Engineering Technology Innovation and Engineering Education Unit) for which the students' professionalization is one of its main objectives. In the current schedule (3 years program), all the courses are given during the end of afternoons (5 pm to 8 pm), with an average of 2 to 3 courses per week. The rest of the day the students are free (most of them are working in local companies). Each course last 15 weeks. The final project or thesis of the students has, in general, no connection with the students' workplace him/her self, though several projects deal with industrial implementations. So, there is no direct connection between the students' companies and the Master program.

In the new proposal, the companies need to be involved in the program in proposing projects and providing internal mentors. The student is either a freshman hired by the

	Sunday	Monday	Tuesday	Wednesday	Thursday
8:00am					
9:00am	Methodological Or Reflexive Sessions (collective)				
12:00pm					
1:00pm	Individual Work on project or meeting with academic tutor	Workplace	Workplace	Workplace	Workplace
3:00pm					
5:00pm	Master courses	Master courses	Master courses	Master courses	Master courses
8:00pm					

**Fig. 1.** Typical weekly schedule for the students

company and provided a salary, or an internal IT engineer or technician (having a bachelor level or equivalent) to be trained within this new cooperative process.

The schedule of the courses will be slightly changed to accommodates the students' needs. The department discusses the projects with the partner companies, and signs an agreement with both the student and the company. The student (becoming a worker at the company) will stay one day every week at the university, the rest of the week he/she works on the project in the company. This university day will focus on the assessment of the students' projects (to be made by the student's mentor), with reflexive sessions where the students exchange their experience and discuss common issues related to their work. It will also deal with methodological sessions where the students are coached on project management issues, and follow seminars and workshops related to specific needs of cybersecurity that the lectures do not cover (social issues, policy issues, etc.). Figure 1 highlights the Typical weekly schedule for the students.

During the students stay at the company, he/she receive the visit of his/her university mentor at his/her workplace several times, and make presentation of his/her achievement in front of his/her both mentors (university mentor, company mentor).

**Table 2.** Cooperative visit planning

Visit	Objectives	Assessment actions
1	Designing a project that is profitable for the company and interesting for student learning	<ul style="list-style-type: none"> <li>• Mutual presentation</li> <li>• Visit of the company</li> <li>• Discussion about the identified projects and their aims</li> <li>• Selection of a project and definition of the working aims for the student</li> </ul>
2		<ul style="list-style-type: none"> <li>• Presentation and validation of the student's project deliverables</li> <li>• 1<sup>st</sup> student assessment (behaviour at work, technical knowledge, communication skills)</li> </ul>
3	Assessing the progressive participation in the project and the involvement in its management	<ul style="list-style-type: none"> <li>• Presentation of the project's progress (used methods, intermediate results, student learning progress)</li> <li>• 2<sup>nd</sup> student assessment (concrete achievements, behaviour at work, technical achievements, management and communication skills)</li> </ul>
4	Assessing the ability of management of the project in progressive autonomy	<ul style="list-style-type: none"> <li>• Presentation of the project's progress (used methods, intermediate results, student learning progress)</li> <li>• 3<sup>rd</sup> student evaluation (concrete achievements, behaviour at work, technical knowledge, management and communication skills)</li> </ul>
5	Finalising the project and thesis writing	<ul style="list-style-type: none"> <li>• Final presentation of the project (used methods, final results, prospective steps within the company)</li> <li>• Discussion of the project thesis document (to be finalised by the student for the Master Jury)</li> <li>• Last student assessment (concrete achievements, behaviour at work, technical knowledge, management and communication skills)</li> </ul>

Table 2 explains the objectives and the tasks of each visit (with an average of one visit per term). Each meeting is specific to the period of the year and to the project's progress. The assessment mainly deals with the concrete achievements of the student, his/her behavior at work and team collaboration, the technical knowledge and the way the student is applying the knowledge he/she acquired at the university, the management of the projects' tasks, the student's communication skills, etc.

### 3.2 Use of Simulation and Ethical Hacking

Beyond their daily project and courses, the students need to continue participating in other specific activities designed for them, such as the department yearly contest on Ethical Hacking. Such contest is an important exercise for the students around Computer Security. They collectively play the role of Ethical Hackers and have the opportunity to understand the weaknesses and vulnerabilities of computer systems to be able to use this knowledge to assess the security robustness of a target system in a lawful and legitimate manner. In general, the ethical hacker exploits open-source tools for forensic analysis and information gathering to disclose encrypted contents. An Ethical Hacker is a professional who uses his knowledge to assess the security robustness of a target system in a lawful and legitimate manner. An ethical hacker can also investigate cybercrimes by exploiting tools for forensic analysis and information gathering to disclose hidden contents ("he is the good guy").

During the contest, the students are mentored to perform basic ethical hacking activities including information gathering, forensic analysis and secrets' disclosure. As an example, the last Ethical Hacking Workshop<sup>4</sup> had three days duration, 2 first days dealing with the "theory and practical applications" and the last day focusing on a "Ethical Hacking contest".

The covered topics during the workshop first days were mainly:

- Ethical Hacking with Linux
- Data hiding techniques
- Information gathering
- Forensic analysis
- Steganography and cryptography.
- Password hacking
- Wireless communication security.
- WiFi security

During the Ethical Contest of the 3rd day, the students were requested to analyze the evidences hidden in a USB stick. The USB stick has been taken from a person involved in a case of industrial secrets stealing. The secrets were related to diagrams and schemes for machineries and equipment in the field of oil and gas extraction and refining. The suspect was able to partially delete the evidences. The students were asked to investigate the USB sticks and identify all the people (name and surname) involved in the case.

---

<sup>4</sup> [http://www.qu.edu.qa/engineering/computer/ethical\\_hacking\\_contest.php](http://www.qu.edu.qa/engineering/computer/ethical_hacking_contest.php).

Each of the evidence discloses information for retrieving the next one. All the evidence files were containing a set of information that should be exploited for the search and identification of the next one.

A presentation by Q-CERT<sup>5</sup> (Qatar Computer Emergency Response Team) was also given during this last day on the policy side, to initiate the students to the national cyber security posture, advices on policies and security standards.

Q-CERT is a national, Government sponsored organization, setup under the auspices of Ministry of Transport & Communications (MOTC). Q-CERT has been instrumental in building resilience into the critical information infrastructure of Qatar and is working to harmonize the secure use of technology through best practices, standard policies, risk mitigations and dissemination of valuable information.

Some demonstrations of Q-CERT projects (Botnet Eradication, Malware Analysis LAB, Threat Monitoring System, etc.) have concluded this day.

Such workshops and contests will be continued all the coming years. The participation of governmental organizations and industrial companies is important. Siemens are Thales groups are already ready to take part in the next workshops.

### 3.3 Tabletop Exercises for Cybersecurity Education

As tabletop exercises are used to give students the opportunity to practice cybersecurity concepts using real world scenarios, this component is also considered. This deals with interactive learning tools and approaches that are based on the idea of training students through role playing on hypothetical problems. The scenarios and the problems are generally derived from real life situations and the method has been successfully used in classroom environment with students in other contexts [6]. Such tabletop approach is being popular and several tabletop exercises are developed by universities and government agencies. For instance, some tabletop templates developed by the Security Operations Center (SOC) of Washington State could be found in [6]. The goal of these templates is to increase the security situational awareness and to facilitate discussion of incident response in as simple a manner possible, targeting a time range of 15 min.

In our case, the scenarios are conducted under the responsibility of the “Cybersecurity Chair”, funded by Thales company (within a Memory of Understanding -MoU- signed between Thales and Qatar University on Nov. 2013)<sup>6</sup>. The “Cybersecurity Chair” is intended to enforce the MoU *“to provide training opportunities for Qatari students on new and emerging technologies in the field of cybersecurity. Its aim is to establish cooperation between the academic environment and industry on information systems and data security and related associated services.”* Hence, the Cybersecurity proof of concepts and scenarios are conducted under the responsibility of the Chair in the offices and labs of Thales, situated in QSTP (Qatar Science Technology Park) in Doha. The design of the scenarios emphasizes on the practice of cooperation, and the familiarization of students to their

---

<sup>5</sup> <http://www.qcert.org/>.

<sup>6</sup> <https://www.thalesgroup.com/en/worldwide/press-release/qatar-university-teams-thales-open-cybersecurity-chair>.

responsibilities, in addition to practice crisis management, test experimental solutions for scenarios, and identify shortcomings in resources, procedures or capabilities.

The Cybersecurity Chair has also implemented a useful package of cybersecurity courses that students have to follow (still under validation by the CSE department):

- Cybersecurity and Cyber Physical Systems
- Cybersecurity in software development
- Cybersecurity in industrial systems
- Cybersecurity Management for Business Managers
- Cybersecurity Management for IT managers/professionals
- SAP Security

The students have also the opportunity to participate in the recently awarded research projects (hereunder) and be part of a cybersecurity experts' network:

- NPRP10-0206-170360: Intrusion Detection System (IDS) for Industrial Control Systems – which uses a systematic study of machine learning schemes to build IDS for Industrial Control Systems. This project aims at building of a Test-Bed environment from oil/gas and petro-chemical industry.
- NPRP10-0105-170107: Cyber Security, Monitoring, Diagnostics, and Resilient Control, Recovery of Cyber-Physical Industrial Control Systems – which deals with the design and development of proactive intrusion attack monitoring and attack resilient control recovery methodologies and toolkits. It mainly aims at building of a simulation environment for Industrial Control Systems.

The main objectives of the tabletop exercises are to explain, in an easy way, the general concepts in cybersecurity, such as the global infrastructure and the discovery process and management of security incidents. Tabletop exercises are an invaluable training tool to prepare the students and train them on practical methods applied in the industry. Some authors such as [7] proposed specific methods that reduce the work load from the instructor perspective, in order to encourage more educators to try the tabletop exercises. There is always a test period where the exercises are applied only to one course and gradually introduce it to other courses. The main benefit of the tabletop method is that it provides scalability to deliver practical experience to a large number of students by one instructor. While our program in Qatar University does not have a real problem of scalability because of the reasonable size of the Master and the use of mentors to implement the internship. Moreover, the students will be exposed to real live systems under the supervision of their company mentors.

In order to provide a more dynamic experience, the exercise format includes a scenario based, but adaptable opponent. In [7], the students are divided into two kinds of teams. Blue teams represent the security or investigative teams of various actors on the defensive side, such as law enforcement, national Computer Emergency Response Team (CERT), Internet Service Provider (ISP), media, industry, etc. The Red team represents a malicious actor (defined by the instructor), such as criminal group, hacktivist organization, hostile intelligence service, etc.

Finally, tabletop instructions should be relatively short and require several teams to fully develop their role in the exercise. The tabletop exercise scenario should cover general cybersecurity concepts and not go into too much technical detail.

## 4 Conclusion

As the country needs to be prepared to protect its infrastructures and sensitive data, contemporary complex issues related to information security and cybersecurity need to be mastered. A strong collaboration between the employment, government, and educational sectors becomes the key for such challenge. This paper highlighted the current initiatives at Qatar University to enforce the Master of Computing curricula, in adding new cooperative tools (Mentorship, Reflexive Sessions, Tabletop methods, Simulation and Ethical Hacking, Specific courses, etc.). International certifications, such as IFIP IP3, ISC2 and SANS for example, tackling new cybersecurity requirements will also bring new issues (risk assessment, threat modeling and design, vulnerability management, etc.) and are gradually integrated within the curricula.

The presented approach is a necessary step before the prospective implementation of a real cooperative program between industry/university based on a sandwich program, where the students gradually develop their skills thanks to their alternate stays between the university and the company (for example 3 weeks in the company and 3 weeks in the university). This needs a serious involvement of all the stakeholders and a substantive change in the Master structure.

**Acknowledgement.** This publication was made possible by NPRP grant # NPRP 7-1883-5-289 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

## References

1. Qatar National Vision 2030. [www.qu.edu.qa/pharmacy/components/upcoming.../Qatar\\_National\\_Vision\\_2030.pdf](http://www.qu.edu.qa/pharmacy/components/upcoming.../Qatar_National_Vision_2030.pdf). Accessed 15 Sept 2017
2. Bouras, A., Veillard, L., Tralongo, S., Lenir, M.: Cooperative education development: towards ICT reference models. In: International Conference on Interactive Collaborative Learning (ICL), pp: 855–861. IEEE Xplore (2014)
3. Kessler, G.C., Ramsay, J.: Paradigms for cybersecurity education in a homeland security program. *J. Homel. Secur. Educ.* **2**, 35–44 (2013)
4. Veillard, L., Tralongo, S., Bouras, A., Le Nir, M., Galli, C.: Designing a competency framework for graduate levels in computing sciences: the Middle-East context. In: Auer, M.E., Guralnick, D., Uhomoihibhi, J. (eds.) ICL 2016. AISC, vol. 544, pp. 330–344. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-50337-0\\_31](https://doi.org/10.1007/978-3-319-50337-0_31)
5. Kessler, G.C., Ramsay, J.D.: A proposed curriculum in cybersecurity education targeting homeland security students. In: 47th Hawaii International Conference on System Science, pp. 4932–4937 (2014)

6. Security Operations Center (SOC), Washington State. <http://soc.wa.gov/node/413>. Accessed 15 Sept 2017
7. Ottis, R.: Light weight tabletop exercise for cybersecurity education. *Homel. Secur. Emerg. Manag.* **11**(4), 579–592 (2014)



# An Energy Saving Mechanism Based on Vacation Queuing Theory in Data Center Networks

Emna Baccour<sup>1(✉)</sup>, Ala Gousssem<sup>1</sup>, Sebti Foufou<sup>2,3</sup>, Ridha Hamila<sup>1</sup>, Zahir Tari<sup>4</sup>, and Albert Y. Zomaya<sup>5</sup>

<sup>1</sup> College of Engineering, Qatar University, Doha, Qatar  
ebaccour@qu.edu.qa

<sup>2</sup> LE2i Lab, University of Burgundy, Dijon, France

<sup>3</sup> Computer Science, New York University, Abu Dhabi, UAE

<sup>4</sup> School of Science, RMIT University, Melbourne, Australia

<sup>5</sup> School of Information Technologies, University of Sydney, Sydney, Australia

**Abstract.** To satisfy the growing need for computing resources, data centers consume a huge amount of power which raises serious concerns regarding the scale of the energy consumption and wastage. One of the important reasons for such energy wastage relates to the redundancies. Redundancies are defined as the backup routing paths and unneeded active ports implemented for the sake of load balancing and fault tolerance. The energy loss may also be caused by the random nature of incoming packets forcing nodes to stay powered on all the times to await for incoming tasks. This paper proposes a re-architecturing of network devices to address energy wastage issue by consolidating the traffic arriving from different interfaces into fewer ports and turning off the idle ones. This paper also proposes to attribute sleeping and active periods to the processing ports to prevent them from remaining active waiting for random arrivals. Finally, we use the vacation queuing theory to model the packets arriving process and calculate the expectation of vacation periods and the energy saved. Then, we strengthen our work with a simulation part that validates the analytical derivations and shows that the proposed mechanism can reduce more than 25% of the energy consumption.

**Keywords:** Power consumption · Vacation queuing theory  
Data center networks

## 1 Introduction

In the last few years, data centers witnessed a revolutionary growth caused by the increasing trend to migrate services, applications, computation and storage into more robust systems. Due to this rapid scaling of data center networks, the unavoidable increase of energy consumption became a challenging problem.

Studies on data center traffic statistics showed that, most of the time, the network operates only at 5% to 25% of its maximum capacity [3, 11] depending

on the period. Also, the network devices have to be always powered on to wait for the unpredictable incoming jobs. However, when the load is low, the idle servers consume 70% of their peak power which causes a great waste of energy [10]. In this context, many research efforts such as ElasticTree [11] tried to make the power consumption proportional to the traffic workload by powering on only the active nodes. These efforts showed interesting results, most of them, however, are based on the traffic load, which is unpredictable. Also, since the traffic load is known to be bursty [11], energy conservation is not always significant especially when the load is high. In addition, defining the set of crucial nodes for the communication and the set of idle nodes to power off is a complex task that has an exponential time.

In order to conceive an energy-aware mechanism independent from the traffic, two facts should be considered. First, duplicating the critical components such as links, ports and servers (also known as *redundancies*) to backup the principal resources in case of failures and insure the network bandwidth may largely contribute in the energy wastage. In fact, maintaining many ports active waiting for packets in a network device can consume a large proportion of power. Although load balancing and fault-tolerance are important especially in heavy traffic loads, it is acceptable to make it optional as most of the time the network does not reach its maximum capacity. Second, the power sleep mode is the key point for energy efficiency. Thus, by limiting the active time of a device and fixing a vacation time, an important amount of energy can usually be saved.

The main contributions of this paper can be summarized as follows:

- Re-architecturing the network devices so that the incoming load is not treated necessarily by its input port but can be relayed to any available processing unit. In fact, we propose to separate the receiving interface from its processing part. In this way, any processing unit can treat the load incoming from any interface. Hence, in low loads, the traffic can be satisfied by only few units and the redundant ports are activated only in need.
- Proposing a packet scheduling algorithm to manage the distribution of tasks and vacation times between different processing units according to their availabilities. In particular, when the buffer of a particular unit is congested, the incoming packets are relayed to the next units. When the buffer is empty, the unit switches to energy efficient mode instead of idle mode.
- Analyzing this approach using the vacation queuing model to expect the vacation period of each unit, the load distribution between units according to the incoming packets, the energy gain compared to the always-on system (system without vacation) and the waiting time of packets in the queues.

The simulation results showed that the proposed approach can achieve the proportionality between the energy consumption and the traffic load. In addition, more than 50% of energy can be reduced in low loads and more than 25% in higher loads while respecting the system performance. Also, compared with the power aware algorithms proposed for data centers, our model owns a much better time and calculation efficiency.

The rest of the paper is organized as follows: Sect. 2 overviews some of the existing works to green data center networks. In Sect. 3, we describe the proposed approach to optimize the energy consumption. An experimental evaluation is provided in Sect. 4 and we conclude the paper in Sect. 5.

## 2 Related Work

Several industrial and academic investigations have been conducted to build a green data center. Some of them chose to use renewable sources of energy instead of brown power such as google data centers [2]. Others suggested to implement power efficient designs such as introducing wireless technology in the data center networks [6]. However, most of the efforts are focusing on saving the energy consumed by idle devices that are not included in the traffic but are still wasting energy. In fact, these works aim to consolidate the arriving traffic flows, restrict the network to a subset of devices and power off the idle nodes. Such approach is studied from different perspectives including routing and queuing perspectives. A short review of related works is summarized as follows:

### 2.1 Routing Level Power Aware Approaches

**ElasticTree.** [11] aimed to find the minimum subset of the network that must be kept active and the set of nodes that are unused and can be shut down. This approach consists of three modules: the optimizer that defines the devices contributing in the traffic communication, the routing module that calculates the packets routes and the power control module which is responsible to adjust the state of devices (on, off). Three optimizers are proposed: the first optimizer aims to find the optimal power saving solution by searching the optimal flow routes while respecting the traffic and performance constraints. However, searching the optimal solution is an NP-hard problem and needs a large computation complexity. Hence, two other optimizers are proposed where the optimal solution is not guaranteed. These optimizers either depend on a known network design or give a non-optimal routing paths to minimize the searching time.

**Vital Nodes Approach.** [5] suggested not to calculate the best routing paths in real time when receiving the traffic pattern. Instead, the network is abstracted to a graph and vital nodes between any two communicating clusters in this graph are calculated using different methods (betweenness, closeness, degree, etc.). These nodes are pre-calculated once when conceiving the network and used directly with a constant computation complexity. At a given time  $t$ , when receiving the traffic matrix, only the vital nodes are kept active.

### 2.2 Queuing Analysis for Power Aware Approaches

Queuing theory is a deeply established analysis that helps to predict the workloads, the performance change, the traffic volumes and scenarios. Few efforts use the queuing models in data centers including:

**Task Managing Based on Vacation M/G/1 Queuing Model.** [8] where the packet scheduling in a data center using an M/G/1 queuing analyze is modeled. The traffic is consolidated into the minimum set of servers and the others are switched off until the receiving of packets. Sejourn time of the packets in the system (time passed in the queue) is calculated and proven to be acceptable while gaining a large amount of energy. Still, since the modeled queue has an unlimited length, real data center scenarios can not be well studied (e.g. congestion, drop rate).

**Task Managing Based on Vacation M/M/n Queuing Model.** [12] proposes a threshold oriented model to reduce the energy consumed by servers in three-tier data centers. Specifically, authors fix an initial number of active servers and power off the others. Then, they keep examining the arriving jobs in the queue. If the queue size reaches a certain threshold, some extra servers must be activated. The optimal trade-off between the power saving and waiting time in the queue is determined by the M/M/n analytical model.

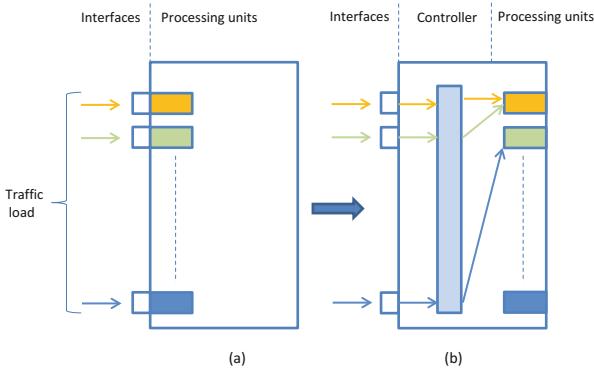
### 3 Proposed Approach

Most of the suggested solutions to green a given network propose to put the interfaces into a lower energy rate depending on the traffic load. In fact, based on the traffic matrix received at an instant  $t$  (servers sending and receiving the communication flows), the set of nodes to power off and the one to keep active are determined. However, the fundamental problem is the unpredictability of the incoming traffic. Also, because of the burst nature of the traffic, the energy is saved only for low loads. To overcome this challenge, we propose to re-architecture the network devices and to use a Sleep/Active algorithm to minimize the dependency on the traffic load and make the energy saving dependent only on the incoming packets at an instant  $t$ . The idea is to attribute vacation periods to the ports that are not receiving any job. To maximize the number of sleeping ports, the interface level (ports receiving the data) is separated from the processing level (units processing the routing decisions), in such a way, the packets received from different interfaces are processed by a minimum number of active units.

#### 3.1 Re-architecturing the Network Devices

The objective of re-architecturing the network devices is that when a number of packets arrive to  $n$  interfaces ( $n$  is the number of ports per device), they can be treated by a smaller number of processing units. In an  $n$ -port device (switch, server,...), each input/output interface is connected to an intelligent unit that processes, extracts and decodes the packet header, looks up to the destination address and decides what to do with it (forward or drop) [9]. The key idea is to decouple each interface from its processing unit. The interfaces level will simply be responsible for receiving, gathering, and forwarding packets to the controller

level. A controller level is implemented to decide what is the available unit to process the packets by respecting the Sleep/Active algorithm described in Sect. 3.2. The Sleep/Active algorithm schedules the distribution of packets among different units based on the queue length and attributes sleeping periods to idle units. The processing unit level is responsible to process, and decode the packets. If the queue of one unit is congested, it notifies the controller to forward the incoming packets to the next unit. In this way, in low loads, the traffic incoming/directed to  $n$  ports can be handled by a lower number of processing units and the idle ones can be turned into sleep status (a) to save considerable amount of energy, (b) to reduce the dependency on the unpredictable load and (c) to reduce the computation complexity. Figure 1(b) shows the proposed re-architected device. Unlike the conventional network device, presented in Fig. 1(a), where every interface has its own processing unit to run routing protocols and decide how to forward packets, routing decisions are stripped from interfaces.



**Fig. 1.** Re-architecturing the network devices.

This new architecture requires new hardware design. In fact, a similar hardware has been proposed and implemented for the Software-Defined Networking (SDN) [13]. An SDN switch separates the data path (packet forwarding) from the control path (routing decisions). The data path portion resides on the switch; a separate controller makes routing decisions. As SDN becomes a trend for cloud computing, the industry is paying more attention to this decoupled hardware including Openflow controllers [1]. Therefore, the proposed re-architected devices can be available to test our approach in a real network.

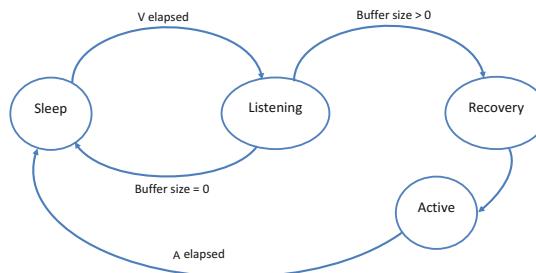
### 3.2 Sleep/Active Algorithm

Given the proposed re-architecturing described above, it now becomes possible to merge packets from multiple interfaces to be processed by few units. However, an algorithm to manage the distribution of tasks and vacation times between

different processing units is important to maximize the sleeping units and, hence, minimize the energy waste.

In the initial stage, all processing units are put into sleep mode. After a period of vacation, say  $V_1$ , their buffers levels are examined. If there are waiting packets, the related processor will be activated, otherwise, it remains in sleeping mode for another period, say  $V_2$ . The first packets communicated through different interfaces are routed automatically to the first processing unit. Whenever a congestion occurs, i.e. buffer level exceeds a congestion window  $K$ , packets will be routed to the adjacent processing unit.

Each processing unit can experience four states: *sleep*, *listening*, *recovery* and *active*. The transition between the *sleep* and *active* state is performed according to the vacation period (say  $V$ ) and the result of the listening period (say  $T_l$ ). When the vacation period elapses, the unit is switched to *listening* state where it examines the buffer status and listens in case of incoming or awaiting packets. If the buffer is empty and there is no appearing traffic load, the processing unit returns to *sleep* state. The *listening* state is performed at the end of every vacation period under a low rate. The passage to the *active* state is triggered when the listening indicates that there are waiting packets. To start serving packets, the unit passes by a recovery period (say  $T_w$ ) where it warms-up between sleeping and active periods. In this paper, the active period lasts for a fixed period (say  $A$ ) and then the unit passes automatically to sleep status, even if the buffer is not empty. The active period consists of multiple service times (times to process  $k$  packets), denoted  $s_1, \dots, s_k$ . Figure 2 provides a summary of the transitions between different status of the processing unit.



**Fig. 2.** Processing unit state diagram

### 3.3 Vacation Queuing Model

In this section, we will analyze our system from a queuing perspective. We will consider an  $M/G/1/K$  queue in which the processing unit goes on vacation for predefined periods after a fixed active period  $A$ . Packets are assumed to arrive according to an independent Poisson process with a rate equal to  $\lambda$  and a distributed queue service time equal to  $\mu$ .

**Expectation of Sleeping Period:** In this part, the expectation of the sleeping period  $S$  for one processing unit is computed. The sleeping period consists of  $N$  vacation periods denoted as  $V_1, \dots, V_N$ , where  $V_N$  is the last vacation period, which means that the listening process reported the arrival of packets. Let  $E_i$  represent the event of no arrivals during the period  $V_i$  and  $E_i^c$  denote the complementary event. To calculate the expectation of the sleeping period  $S$ , we need first to compute the distribution of the number of vacations  $N$ . The expected value of  $N$  is given by the following expression:

$$E[N] = \sum_{i=0}^{\infty} i P(N = i). \quad (1)$$

Once we calculate the distribution of  $N$ , we can compute the expectation of  $S$  which is composed of  $N$  vacation periods  $(V_1, \dots, V_N)$ . In this context, since we are dealing with a non-negative random value  $V_i$ , we can use the Laplace Stieltjes transform of  $V_i$  which is very useful to simplify calculations in the applied probabilities and queuing theory. The Laplace Stieltjes transform of  $V_i$  can be written as  $L_{V_i}(s) = E[e^{-sV_i}]$ . The probability of having a certain number of vacations can be calculated as follows:

$$\begin{aligned} P(N = 1) &= P(E_1^c) = E[1 - e^{-\lambda V_1}] = 1 - L_{V_1}(\lambda). \\ P(N = i) &= P(E_1).P(E_2).P(E_3) \dots P(E_i^c) = (1 - L_{V_i}(\lambda)).\prod_{j=1}^{i-1} P(E_j) \\ &= (1 - L_{V_i}(\lambda)).\prod_{j=1}^{i-1} L_{V_j}(\lambda). \\ P(N \geq i) &= P(E_1).P(E_2).P(E_3) \dots P(E_{i-1}) = \prod_{j=1}^{i-1} P(E_j) = \prod_{j=1}^{i-1} L_{V_j}(\lambda). \end{aligned} \quad (2)$$

Using Eq. (1), the expected *number of vacations* can be defined as:

$$E[N] = \sum_{i=0}^{\infty} i P(N = i) = \sum_{i=0}^{\infty} P(N \geq i) = \sum_{i=0}^{\infty} \prod_{j=1}^{i-1} L_{V_j}(\lambda). \quad (3)$$

We assume that the vacation periods are mutually independent and only depend on the no arrival of packets in the listening period. Hence, the expectation of the sleeping period can be calculated as follows:

$$S = \sum_{i=1}^N V_i = \sum_{i=1}^{\infty} V_i \mathbb{1}_{\{N \geq i\}}, \quad E[S] = \sum_{i=1}^{\infty} E[V_i] \prod_{j=1}^{i-1} L_{V_j}(\lambda). \quad (4)$$

Where,  $\mathbb{1}\{N \geq i\}$  is equal to 1 when  $N \geq i$  and 0 when  $N < i$ .

**Distribution of Load Between Units:** Let  $P_B$  denote the probability to relay the packet to the next unit, if the queue is full. Consequently,  $(1 - P_B)$  represents the probability that an arrived packet is accepted.  $\lambda_U$  denotes the effective data rate of the system which is the number of packets that are actually served by the unit. We introduce also the offered load  $\rho = \lambda\mu$  defined by Little's law [4] and similarly the effective carried load denoted by  $\rho_U$ .  $\lambda_U$  and  $\rho_U$  are given respectively by  $\lambda(1 - P_B)$  and  $\rho(1 - P_B)$ . Hence, the probability  $P_B$  can be written as  $P_B = \frac{\rho - \rho_U}{\rho}$ .

The *effective carried load*  $\rho_U$  is also defined as the probability that the unit is busy at an arbitrary time in a long period of time  $P$  [7].  $\rho_U$  can be written:

$$\rho_U = \lim_{P \rightarrow \infty} \frac{\sum \text{service periods in } P}{\sum \text{vacation periods in } P + \sum \text{service periods in } P} = \frac{P(E_s)\mu}{P(E_v)\bar{v} + P(E_s)\mu}, \quad (5)$$

where  $\bar{v}$  is the mean vacation time,  $E_s$  is the event of being in the end of service and  $E_v$  is the event of being in the end of vacation which is expressed by:

$$P(E_v) = P(E_v^v) + P(E_v^s) = P(E_v^v) + P(E_v|E_s)P(E_s) = P(E_v^v) + \frac{1}{A}P(E_s),$$

where  $E_v^v$  is the event to pass from a vacation to another one and  $E_v^s$  is the event to pass from service to vacation. Since we know that  $P(E_v) + P(E_s) = 1$ , we can deduce  $P(E_v)$  and  $\rho_U$ :

$$P(E_v) = \frac{P(E_v^v) + 1/A}{1/A + 1}, \rho_U = \frac{(1 - P(E_v^v))\mu}{(P(E_v^v) + 1/A)\bar{v} + (1 - P(E_v^v))\mu}. \quad (6)$$

To evaluate probability of passing from vacation to another  $P(E_v^v)$ , we will use the imbedded Markov Chain approach [7]. This approach is widely recognized as a powerful tool for the study of queues. It helps to predict the state of the unit queue at a random time  $t$  and to define the performance of the system (waiting time in the queue). The key idea of Markov Chain approach is to choose random points and calculate the probabilities of being in active or sleeping periods.

Markov points are chosen randomly from time instants when a vacation time ends or a service time ends. We assume here that the recovery period is small which is not enough to receive packets. We will consider the following probabilities:

- $q_k$ : the probability of  $k$  jobs waiting when the vacation period ends ( $k = 0, 1, \dots, K$ ); Note that  $q_0$  is equal to  $P(E_v^v)$ .
- $\pi_k$ : the probability of  $k$  jobs waiting when the service period ends ( $k = 0, 1, \dots, K-1$ ); Note that after a service time, the queue size cannot be  $K$  because at least one packet was treated.
- $f_j$ : the probability of receiving exactly  $j$  arrivals during a vacation period ( $j = 0, 1, \dots, \infty$ ).
- $a_j$ : the probability of receiving exactly  $j$  arrivals during the service period ( $j = 0, 1, \dots, \infty$ ).

Since arrivals are assumed to form a Poisson process, we have:

$$f_j = \int_0^\infty \frac{(\lambda t)^j}{j!} e^{-\lambda t} v(t) dt, \quad a_j = \int_0^\infty \frac{(\lambda t)^j}{j!} e^{-\lambda t} s(t) dt, \quad (7)$$

where  $v(t)$  is the probability density function of the vacation periods with a mean  $\bar{v}$  and  $s(t)$  is the probability density function of the service periods with a mean  $\frac{1}{\mu}$ . After an imbedded point, the system state can be defined as follows:

$$q_k = q_0 f_k + \left( \sum_{j=0}^k f_{k-j} \right) P(E_v). \quad k = 0, 1, \dots, (K-1). \quad (8)$$

$$q_K = q_0 \sum_{k=K}^{\infty} f_k + \left( \sum_{k=K}^{\infty} \sum_{j=0}^k \pi_j f_{k-j} \right) P(E_v). \quad k = K. \quad (9)$$

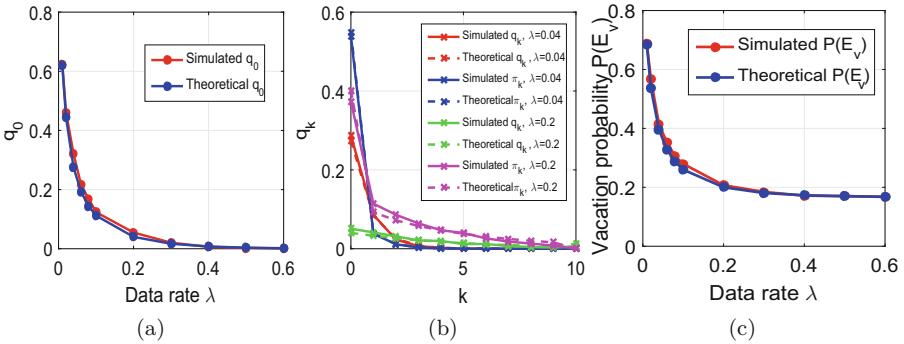
$$\pi_0 = q_1 a_0 + (1 - P(E_v))(p_0 a_0 + p_0 a_0 + p_0 a_1 + p_1 a_0). \quad (10)$$

$$\pi_k = \sum_{j=1}^{k+1} q_j a_{k-j+1} + \left( \sum_{j=0}^{k+1} \pi_j a_{k-j+1} \right) (1 - P(E_v)). \quad k = 1, \dots, (K-2). \quad (11)$$

$$\pi_{K-1} = \sum_{j=1}^k q_j + \left( \sum_{j=0}^{k-1} \pi_j \sum_{k=K}^{\infty} a_{k-j+1} \right) (1 - P(E_v)). \quad k = K-1. \quad (12)$$

Since  $\sum_{k=0}^K q_k$  and  $\sum_{k=0}^{K-1} \pi_k$  are complementary, we have also  $\sum_{k=0}^K q_k + \sum_{k=0}^{K-1} \pi_k = 1$ .

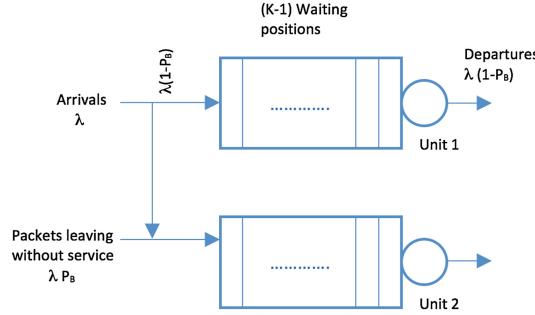
This system is solved using CVX toolbox of Matlab to compute all probabilities. To validate our theoretical derivations, we compared them to the simulated arrival process and Sleep/Active mechanism with the same parameters. Figures 3(a), (b) and (c) show that the theoretical derivations of  $q_0$ ,  $q_k$ ,  $\pi_k$  and  $P(E_v)$  are very close to the simulation.



**Fig. 3.** Comparison between simulated and theoretical probabilities.

Until now, we analyzed the system  $M/G/1/K$  for a single unit queue with a Poisson arrival process. However, a network machine has multiple processing units. Therefore, we need to generalize our study to an  $M/G/n/K$  system, where  $n$  is the number of units per network device. As shown in Fig. 4, at a random time  $t$ , the unit can process only one packet and its queue has  $(K-1)$  waiting positions, where the jobs can wait if they find the unit busy on arrival. This queue can have  $K$  waiting positions if the unit is on vacation. Packets arriving when the system is full are not allowed to enter the queue and will be relayed to the next unit.

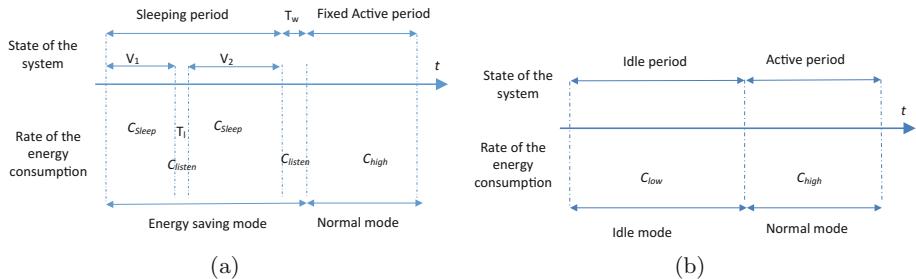
Only a fraction  $(1 - P_B)$  of the arrivals actually enters the queue of the first unit. The *effective arrival rate* of packets waiting in the queue is only  $\lambda_U =$



**Fig. 4.** Distribution of effective data rate between units.

$\lambda(1 - P_B)$ . Each unit experiences the rejection mechanism and relays the data to the next unit. Therefore, the traffic load is distributed among units with different rates. In this work, we assume that the blocked load leaving the first unit and entering the next unit follows a Poisson process with a parameter  $\lambda P_{B_1}$ . Following this assumption, we can write:  $\lambda_{U_1} = (1 - P_{B_1})$  and  $\lambda_{U_i} = \lambda \prod_{j=1}^{i-1} P_{B_j} (1 - P_{B_i})$ , where  $\lambda_{U_i}$  is the *effective rate* of the  $i^{\text{th}}$  unit; ( $i = 2 \dots n$ ).

**Expectation of Energy Gain:** To calculate the energy gain, we will compare the proposed system described in Fig. 5(a) to the always-on device presented in Fig. 5(b) (where conventional network devices are used and vacation queuing model is not applied).



**Fig. 5.** Comparison between power-aware system and always-on system.

Since a processing unit can be in different states (i.e. *sleeping*, *active*, *listening* and *recovery*), we can distinguish between the following three possible energy levels which are from the highest to the lowest:  $C_{high}$  consumed during the process of packets (active period  $A$ );  $C_{listen}$  experienced when checking the state of the queue (listening period  $T_l$ ) or in the recovery period  $T_w$ ; and  $C_{sleep}$  consumed when the unit is inactive (sleeping period  $E[S]$ ).

During sleeping period, we observe that there are  $E[N]$  listening periods and one recovery period  $T_w$ . The energy consumption of the whole system per time unit can be defined as follows:

$$E_{Power-aware} = \frac{A}{(E[S] + T_w + A + T_l E[N])} C_{high} + \frac{(T_l E[N] + T_w) C_{listen}}{(E[S] + T_w + A + T_l E[N])} (13)$$

$$+ \frac{E[S] C_{sleep}}{(E[S] + T_w + A + T_l E[N])}.$$

If the power saving mechanism is not active (always-on system), the power consumption when there is a load is equal to  $C_{high}$  and it is equal to  $C_{low}$  in idle periods. So, its energy consumption can be calculated as follows:

$$E_{Always-on} = \rho C_{high} + (1 - \rho) C_{low}.$$

The economy of energy when using the power-aware mechanism comparing to the always-on mechanism is equal to  $(E_{Always-on} - E_{Power-aware})$ . Thus, we can define the relative energy gain as:

$$EG = \frac{E_{Always-on} - E_{Power-aware}}{E_{Always-on}}. \quad (14)$$

The mean energy gain per port of a network device is the average of energy gains of its units.

## 4 System Evaluation

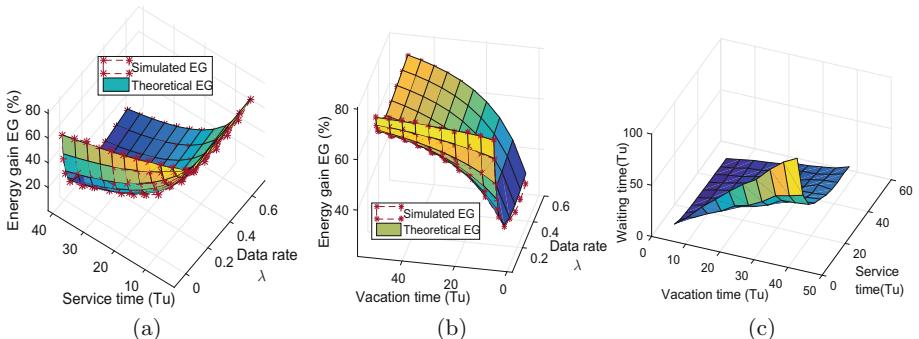
Based on the metrics estimated in the Sect. 3.3 and by simulating the arriving process and the proposed queuing model on Matlab, we can calculate the efficiency and the performance of our system. But, first, some configurations need to be defined. Let  $v(t)$  and  $s(t)$  (the probability density functions of the vacation periods and service periods respectively) have an exponential distribution. We fix, then, the packet service time  $\mu = 1\text{Tu}$  (Time unit),  $T_w = T_l = 0.5\text{Tu}$  and the maximum number of packets in the queue  $K = 10$ . The mean duration of each vacation time is equal to  $\bar{v}_i$ ; hence its related Laplace transform can be written as indicated in (15). In our simulation, we consider that all vacation times are equal and their mean size is equal to  $\bar{v}$ . In this way, the expression of  $E[N]$  and  $E[S]$  can be simplified as follows:

$$\begin{cases} E[V_i] = \bar{v}_i & i = 1, 2, \dots \\ L_{V_i}(\lambda) = \frac{1}{1 + \bar{v}_i \lambda} & i = 1, 2, \dots \end{cases} \Rightarrow \begin{cases} E[N] = 1 + \frac{1}{\bar{v} \lambda} \\ E[S] = \bar{v} + \frac{1}{\lambda} \end{cases} \quad (15)$$

The values of the energy consumed in different states are:  $C_{high} = 5000^3 \times 10^{-6}$  watts;  $C_{low} = C_{high} \times 0.7$ ;  $C_{listen} = C_{high} \times 0.3$ ;  $C_{sleep} = C_{high} \times 0.1$ .

As a first step, we compared our theoretical results and simulation results and we proved the integrity of our calculations as we can see in Fig. 6. These results show that the network packets arrival rate has a great impact on the

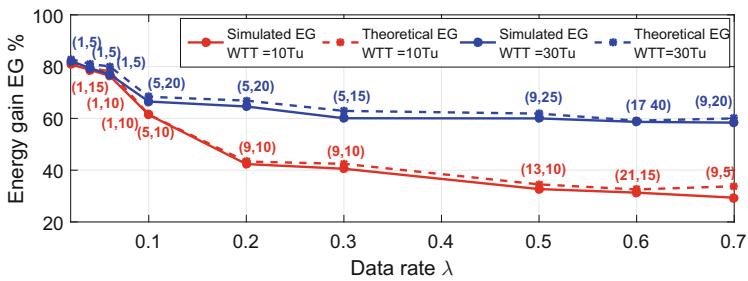
energy consumption. As shown in Figs. 6(a) and (b), when increasing the network load, the power consumption of one processing unit increases and the power conservation degrades for different sizes of active period  $A$  and vacation time  $\bar{v}$ . This means that our system accomplishes more power saving at the lowest network rate. In addition, in the higher rates, the energy consumed is quasi-constant and the energy gain is always important. Our important energy gain in high loads is achieved due to the fixed active period imposed to the processing units and the switching to sleeping state even if the queue is not empty. In this way, our power-aware system achieves more than 15% of energy gain. The energy gain is also dependent of two other parameters which are the duration of the service period and the vacation time. As we can see in Fig. 6(a), the size of  $A$  has an impact on the energy for different values of  $\lambda$  and a mean vacation time equal to 10 Tu. More precisely,  $EG$  decreases by the increase of  $A$ . The contribution of  $A$  is more visible in high loads because in low loads the arrivals occur seldomly. So, the number of vacation periods is big and the service period is always small comparing to it. However, the sleeping period in high loads, generally, consists only of one vacation period which explains the bigger impact of  $A$  for these high rates. Figure 6(b) presents the impact of  $\bar{v}$  on the energy gain when  $A$  is equal to 10 Tu. We can see that the energy increases greatly and monotonically with the increase of the mean vacation especially in high loads. The great impact of the vacation in high loads is explained by the fact that the energy is always high in very low loads due to the large number of vacation times which is not the case of higher loads composed generally of one vacation and one small service ( $A = 10$  Tu). So, by increasing  $\bar{v}$ ,  $A$  becomes insignificant.



**Fig. 6.** Impact of the service and vacation on the energy gain and waiting time.

Therefore, to gain the maximum of energy, the active period  $A$  should be minimized and the mean vacation  $\bar{v}$  should be maximized. However, the performance of the network should always be considered which is in our system the waiting time in the queue. Figure 6(c) presents the impact of the duration of vacation and service on the waiting time when  $\lambda = 0.2$ . We can see that the waiting time is smaller when the service is bigger and the vacation duration is

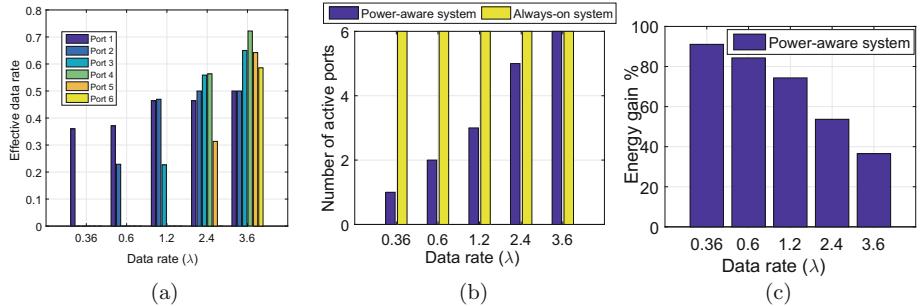
minimized. Since the maximization of the energy gain and the maintain of the system performance are in contrast, a trade off should be established. A waiting time threshold (WTT) should be chosen and the energy can be maximized while respecting the constraint. This constraint is chosen by the owner of the applications implemented in the data center as he knows the performance requirements of his services. Figure 7 presents the energy gain when fixing two WTT thresholds equal to 10 Tu and 30 Tu and choosing  $(A, \bar{v})$  that maximize the energy. We can see that a bigger threshold contributes to gain more energy. Hence, the applications owners can sacrifice a little bit in terms of latency which is impacted by the waiting time in order to reduce the budget of the energy. Theoretical and simulated results are proved to be aligned.



**Fig. 7.** Energy gain for different waiting time thresholds.

As explained in the previous section, after the re-architecturing of the network devices, the first processing units will accept higher load rates. However, the rest of the units will switch to sleep state or accept only low packets rates. Figure 8(a) shows the effective data rate processed by each unit in the new re-architectured device. In fact, the data rate  $\lambda$  is equal to the sum of all loads received by all the interfaces of the network device ( $n=6$ ). For example, when  $\lambda = 1.2$ , this means that each interface received a load equal to 0.2. As we can see, for lower loads, the received data is processed by only 2 or 3 units (see Fig. 8(b)). Hence, our approach proved its efficiency to maximize the number of sleeping units and save energy even in high loads, which is depicted in Fig. 8(c) without being dependent on the traffic matrix. The size of vacation and service must be well optimized while respecting the performance requirements of a data center to ensure better results and a good distribution of load between units (blocking probability depends on  $A$  and  $\bar{v}$ ) which will be studied in future works in addition to the implementation of the solution in a data center architecture.

To conclude, comparing to the power aware algorithms described in the Sect. 2, the proposed power saving approach can be applied to any data center topology and it is not dependent on the architecture of the network since we only change in the network devices and not the interconnection. In addition, one of the biggest advantages of our re-architecturing and queuing approach



**Fig. 8.** Load distribution among different units.

lies in its negligible calculation complexity. Generally, the power aware routing algorithms have a bad exponential time complexity due to the huge searching space for communication flows routes. However, our approach is just relaying the incoming packets one by one to the available processing unit without any calculation complexity.

## 5 Conclusion

In this paper, we studied the idea of decoupling the network device interfaces from their processing units. In this way, the incoming loads from different interfaces can be handled by only few available units. The other ones will be switched into sleep state. To maximize the number of sleeping units and manage the distribution of incoming packets, a Sleep/Active algorithm is proposed. Then, an analysis following the M/G/1/K queuing model is conducted to estimate the energy gain and the sleeping periods. The simulation results aligned with the theoretical study proved the efficiency of the system.

**Acknowledgments.** This paper was made possible by NPRP grant 6-718-2-298 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

## References

1. Openflow (2011). <http://archive.openflow.org/>
2. Google green (2017). <https://environment.google/>
3. Abts, D., Marty, M.R., Wells, P.M., Klausler, P., Liu, H.: Energy proportional datacenter networks. In: Proceedings of the 37th Annual International Symposium on Computer Architecture, pp. 338–347. ACM (2010)
4. Adan, I., Resing, J.: Queueing Systems (2015)
5. Baccour, E., Foufou, S., Hamila, R., Tari, Z.: Achieving energy efficiency in data centers with a performance-guaranteed power aware routing. Comput. Commun. **109**, 131–145 (2017)

6. Baccour, E., Foufou, S., Hamila, R., Hamdi, M.: A survey of wireless data center network. In: Proceedings of CISS-15 International Conference on Information Sciences and Systems March 2015
7. Bose, S.K.: An Introduction to Queueing Systems. Heidelberg, Springer (2002)
8. Cheng, C., Li, J., Wang, Y.: An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing. *Tsinghua Sci. Technol.* **20**, 28–39 (2015)
9. Froom, R., Frahim, E.: Implementing Cisco IP Switched Networks (SWITCH) Foundation Learning Guide: (CCNP SWITCH 300–115). Pearson Education, Indianapolis (2015)
10. Greenberg, A., Hamilton, J., Maltz, D.A., Patel, P.: The cost of a cloud: research problems in data center networks. *SIGCOMM Comput. Commun.* **39**, 68–73 (2008)
11. Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S., McKeown, N.: ElasticTree: saving energy in data center networks. In: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, p. 17 (2010)
12. Schwartz, C., Pries, R., Tran-Gia, P.: A queuing analysis of an energy-saving mechanism in data centers. In: The International Conference on Information Network, pp. 70–75 (2012)
13. Xia, W., Wen, Y., Foh, C.H., Niyato, D., Xie, H.: A survey on software-defined networking, pp. 27–51 (2015)



# Homomorphic Evaluation of Database Queries

Hamid Usefi<sup>1,2(✉)</sup> and Sudharaka Palamakumbura<sup>1</sup>

<sup>1</sup> Department of Mathematics and Statistics, Memorial University of Newfoundland,  
St. John's, NL A1C 5S7, Canada  
[{usefi,sudharakap}@mun.ca](mailto:{usefi,sudharakap}@mun.ca)

<sup>2</sup> Department of Mathematics and Computer Science,  
AmirKabir University of Technology, 424 Hafez Avenue, 159163-4311 Tehran, Iran

**Abstract.** Homomorphic encryption is an encryption method that enables computing over encrypted data. This has a wide range of real world ramifications such as being able to blindly compute a search result sent to a remote server without revealing its content. This paper discusses how database search queries can be made secure using a homomorphic encryption scheme. We propose a new database search technique that can be used with the ring-based fully homomorphic encryption scheme proposed by Braserski.

**Keywords:** Homomorphic · Privacy · Encryption · Database · Query

## 1 Introduction

In this work, we address the problem of searching privately on a database. We consider the case that the database is stored on a third-party server. To protect the data, we first encrypt the database and then upload the encrypted database on an untrusted server and the server does not have access to our secret key. Now we want to send a search request to the server. Since the data on the server is encrypted, we would need to search on ciphertexts and output a ciphertext. We can then decrypt the ciphertext. This scheme would work only if we are able to do computations on ciphertexts. The adaptation of these kind of services in health care are becoming increasingly common with cloud-based health recording and genomic data management tools such as Microsoft Health.

Homomorphic encryption allows computations to be carried out on the cipher-text such that after decryption, the result would be the same as carrying out identical computations on the plain-text. This has novel implications such as being able to carry out operations on database queries in the form of

---

H. Usefi—The research is supported by NSERC of Canada under grant # RGPIN 418201 and the Research & Development Corporation of Newfoundland and Labrador.

cipher-text and returning the result to the user so that no information about the query is revealed at the server's end [7].

In this paper, we employ homomorphic encryption (HE). The idea of homomorphic encryptions is not new, and even the oldest of ciphers, ROT13 developed in ancient Rome, had homomorphic properties with respect to string concatenations [5]. Certain modern ciphers such as RSA and El Gamal also support homomorphic multiplication of cipher texts [5].

The idea of a “fully” homomorphic encryption scheme (or *privacy homomorphism*) which supports two homomorphic operations was first introduced by Rivest et al. in [4]. After more than three decades, the first fully homomorphic encryption scheme was founded by Gentry in 2009 with his breakthrough construction of a lattice based cryptosystem that supports both homomorphic additions and multiplications [6]. Although the lattice based system is not used in practice, it paved the way for many other simpler and more efficient fully homomorphic models constructed afterwards.

At a high level, Gentry's idea can be described by the following general model. This is the blueprint that is used in all homomorphic encryption schemes that followed.

1. Develop a *Somewhat Homomorphic Encryption Scheme* that is restricted to evaluating a finite number of additions or multiplications.
2. Modify the somewhat homomorphic encryption scheme to make it *Bootstrapable*, that is, modifying it so that it could evaluate its own decryption circuit plus at least one additional NAND gate.

Every probabilistic encryption function usually introduces a *noise* and when the noise exceeds a certain threshold, the decryption function does not return the desired plain-text. The idea behind constructing a bootstrappable scheme is that whenever the noise level is about to reach the threshold, we can *refresh* the cipher-text and get a new cipher-text so that these cipher-texts decrypt to the same pain-text but the new cipher-text will have a lower noise. In this way, if the cipher-text is bootstrapped from time to time, an arbitrary number of operations can be carried out.

Gahi et al. [1] use homomorphic encryption to search for an encrypted message on a database that is not encrypted. Their work specifically uses the DGHV fully homomorphic encryption scheme [2]. The DGHV scheme operates on plain-text bits separately, and thus Gahi's method requires a large amount of computations to perform even on a simple operation such as integer multiplication. We view a database as an  $\ell \times n$  matrix, where  $\ell$  is the number of records (patients). Now corresponding to each record, there are  $n$  lab results (columns). In Sect. 5.1, we shall first address the type of queries whose return output is a unique record. For example, we want to search for a record with a given ID number, that is one of the columns in our matrix corresponds to ID numbers and in that column we want to search for a specific ID number. We use a more modern fully homomorphic encryption schemes. In particular, we modify recent ring based fully homomorphic encryption scheme proposed by Brakerski and Vaikuntanathan [3] which work on blocks of data rather than single bits (as in Gahi's scheme).

As such the number of computations can be greatly reduced. Next we extend our method to a more general setting. To this end, we use a recent result of Kim et al. [10] where they considered equality testing on the cipher texts. Suppose we have a function EQUAL that as input takes two cipher texts  $c_1$  and  $c_2$ , where  $c_1 = \text{Enc}(m_1)$  and  $c_2 = \text{Enc}(m_2)$ . The output of EQUAL is  $\text{Enc}(1)$  if  $m_1 = m_2$  and  $\text{Enc}(0)$ , otherwise. Our general case, discussed in Sect. 5.3, addresses searches for a keyword with exact matching. We are able to search over a column of the database. Let  $R_{i,j}$  be the attribute of record  $i$  in column  $j$ . Given a keyword  $m$ , we return those records  $R_i$  such that  $\text{Enc}(m) = \text{Enc}(R_{i,j})$ . This can be easily extended for multiple keyword search with exact matching.

## 2 DGHV Fully Homomorphic Encryption

The DGHV scheme was introduced by Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan in 2010, and this scheme operates on integers as opposed to lattices in Gentry's original construction. The scheme follows Gentry's original blueprint by first constructing a somewhat homomorphic encryption scheme. The key generation, encryption and decryption algorithms of the DGHV scheme are given below.

Let  $\lambda \in \mathbb{N}$  be the security parameter and set  $N = \lambda$ ,  $P = \lambda^2$  and  $Q = \lambda^5$ . The scheme is based on the following algorithms;

- **KeyGen( $\lambda$ )**: The key generation algorithm that randomly chooses a  $P$ -bit integer  $p$  as the secret key.
- **Enc( $m, p$ )**: The bit  $m \in \{0, 1\}$  is encrypted by

$$c = m' + pq,$$

where  $m' \equiv m \pmod{2}$  and  $q, m'$  are random  $Q$ -bit and  $N$ -bit numbers, respectively. Note that we can also write the cipher-text as  $c = m + 2r + pq$  since  $m' = m + 2r$  for some  $r \in \mathbb{Z}$ .

- **Dec( $c, p$ )**: Output  $(c \bmod p) \bmod 2$  where  $(c \bmod p)$  is the integer  $c'$  in  $(-p/2, p/2)$  such that  $p$  divides  $c - c'$ .

The value  $m'$  is called the *noise* of the cipher-text. Note that this scheme, as it is given above, is symmetric (i.e., it only has a private key). We can define the public key as a random subset sum of encryptions of zeros, that is, the public key is a randomly chosen sum from a predefined set of encryptions of zeros:  $S = \{2r_1 + pq_1, 2r_2 + pq_2, \dots, 2r_n + pq_n\}$ . A typical encryption of the plain-text  $m$  would be,

$$\begin{aligned} c &= m + \sum_{i \in T} (2r_i + pq_i) \\ &= m + 2 \sum_{i \in T} r_i + p \sum_{i \in T} q_i, \end{aligned}$$

where  $T \subseteq S$ . From here on we shall use  $m'$  to denote  $m + \sum_{i \in T} r_i$  and  $q$  to denote  $\sum_{i \in T} q_i$ .

This scheme is homomorphic with respect to addition and multiplication and decrypts correctly as long as the noise level does not exceed  $p/2$  in absolute value. That is,  $|m'| < p/2$ . Hence, this is a somewhat homomorphic encryption scheme in the sense that once the noise level exceeds  $p/2$ , the scheme loses its homomorphic ability. It is shown that this scheme is Bootstrappable.

### 3 Query Processing Using the DGHV Scheme

The DGHV scheme can be used to create a protocol that establishes blind searching in databases. This method was proposed by Gah et al. [1].

Suppose we need to retrieve a particular record from the database. Typically, we send a query to the database encrypted using the DGHV scheme. Let  $v_i$  be the  $i^{\text{th}}$  bit of the query  $v$  and  $c_i$  be the  $i^{\text{th}}$  bit of a record  $R$  in database  $D$ . Both the query and the database record is encrypted using the DGHV scheme. Suppose the plain-text bit corresponding to  $v_i$  is  $m_i$  and the plain-text bit corresponding to  $c_i$  is  $m'_i$ . Then,

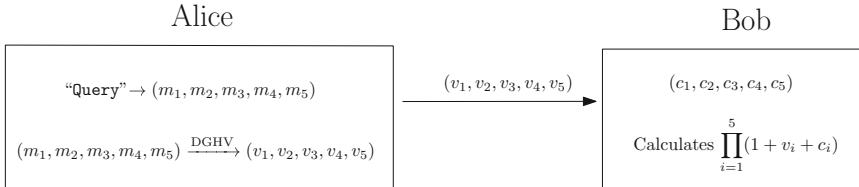
$$v_i = m_i + 2r_i + pq_i$$

and

$$c_i = m'_i + 2r'_i + pq'_i,$$

where  $r_i, r'_i, q_i$  and  $q'_i$  are random numbers and  $p$  is the secret key. The server shall compute the following sum for each record  $R_t$  with index  $t$ :

$$I_t = \prod_i (1 + c_i + v_i). \quad (1)$$



**Fig. 1.** Calculation of  $I_r$  values.

We observe that

$$1 + c_i + v_i = 1 + (m_i + m'_i) + 2(r_i + r'_i) + p(q_i + q'_i).$$

So, if  $m_i = m'_i$ , then  $m_i + m'_i \equiv 0 \pmod{2}$ . In this case:

$$1 + c_i + v_i = \text{Enc}(1).$$

On the other hand, if  $m_i \neq m'_i$ , then  $m_i + m'_i \equiv 1 \pmod{2}$ . Therefore,

$$\begin{aligned} 1 + c_i + v_i &= 2(1 + r_i + r'_i) + p(q_i + q'_i) \\ &= \text{Enc}(0). \end{aligned}$$

This results in  $I_t = \text{Enc}(0)$ . Hence, for each record  $R_t$  in the database we will have an  $I_t$  value that is equal to  $\text{Enc}(1)$  or  $\text{Enc}(0)$  depending on whether the search query  $m$  matches  $R_t$  or not (Fig. 1).

Next, we calculate the partial sums of the  $I_t$  values:

$$S_r = \sum_{t \leq r} I_t. \quad (2)$$

As an example, let us consider a database that has five records, each encoded with 4 bits. If the query sent by the user is  $(\text{Enc}(1), \text{Enc}(1), \text{Enc}(0), \text{Enc}(0))$ , we obtain the corresponding  $I_r$  and  $S_r$  values, as shown in Table 1.

**Table 1.** Sample database with corresponding  $I_r$  and  $S_r$  values

Database records	$I_r$	$S_r$
$(1, 1, 0, 0)$	$\text{Enc}(1)$	$\text{Enc}(1)$
$(1, 0, 1, 0)$	$\text{Enc}(0)$	$\text{Enc}(1)$
$(1, 1, 0, 0)$	$\text{Enc}(1)$	$\text{Enc}(2)$
$(1, 1, 0, 1)$	$\text{Enc}(0)$	$\text{Enc}(2)$
$(1, 0, 0, 0)$	$\text{Enc}(0)$	$\text{Enc}(2)$

Next, we calculate the sequence  $I'_r = (I'_{r,j})$  for every record  $R_r$  with index  $r$  and every positive integer  $j \leq r$ :

$$I'_{r,j} = I_r \prod_i (1 + \bar{j}_i + S_{r,i}), \quad (3)$$

where  $S_{r,i}$  is the  $i^{\text{th}}$  bit of  $S_r$  and  $\bar{j}_i$  represents the  $i^{\text{th}}$  bit of the encryption of  $j$ . Hence, these sequences have the property that whenever  $I_r = \text{Enc}(1)$  and  $S_r = \text{Enc}(j)$ , we have  $I'_{r,j} = \text{Enc}(1)$ . Otherwise,  $I'_{r,j} = \text{Enc}(0)$ . Following the example given in Table 1, we get

$$\begin{aligned} I'_1 &= (\text{Enc}(1)), \\ I'_2 &= (\text{Enc}(0), \text{Enc}(0)), \\ I'_3 &= (\text{Enc}(0), \text{Enc}(1), \text{Enc}(0)), \\ I'_4 &= (\text{Enc}(0), \text{Enc}(0), \text{Enc}(0), \text{Enc}(0)), \\ I'_5 &= (\text{Enc}(0), \text{Enc}(0), \text{Enc}(0), \text{Enc}(0), \text{Enc}(0)). \end{aligned}$$

Finally, we calculate the sequence

$$R' = \sum_k \text{Enc}(R_k) I'_k, \quad (4)$$

where  $R_k$  is the  $k^{\text{th}}$  record in  $D$ . So,  $R'$  is a sequence containing only the encrypted records that matches our search query. Note that the definition of

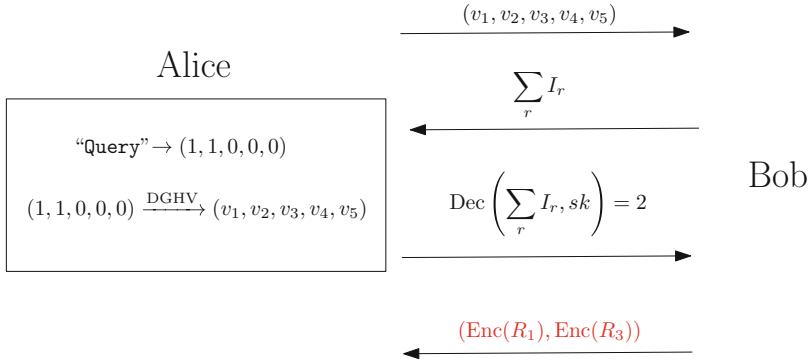
$R'$  relies on adding vectors of different lengths. This is done in the natural way, whereby all the vectors are made the same length by padding with zeros prior to addition. In the above example, we obtain,

$$R' = (\text{Enc}(R_1), \text{Enc}(R_3), \text{Enc}(0), \text{Enc}(0)).$$

At this point, the sequence  $R'$  will contain all the records that match our query, but with trailing encryptions of zeros we do not need. Hence, a second sum is calculated at the server side to determine the number of terms that are useful in the sequence:

$$n = \sum_r I_r$$

This result can be returned to the user and decrypted to obtain the number of records that match the search query. Hence, the sequence  $R'$  can be truncated at the appropriate point and returned to the user for decryption. The whole process is illustrated in Fig. 2.



**Fig. 2.** Alice, Bob, and Gahi’s protocol.

An update query can be performed by,

$$R_{new} = (1 + I_r)R + I_r U, \text{ for every } R \in D,$$

where  $U$  is the new value that we wish to insert whenever the query matches  $R$  (or  $I_r = \text{Enc}(1)$ ). A deletion of a record can be performed by,

$$R_{new} = (1 + I_r)R \text{ for every } R \in D.$$

To perform all these operations without exceeding the maximum noise permitted ( $p/2$ ), it is necessary to choose the parameters  $N$ ,  $P$ , and  $Q$  appropriately.

Gahi’s method works on plain-text bits and thus requires significant computational ability on the part of the server. This is due to the fact that it is restricted to the DGHV scheme which processes plain-text bits separately. Now we propose an alternative protocol called the Homomorphic Query Processing Scheme. This protocol enables us to process database queries using more modern fully homomorphic encryption schemes such as the ring based scheme proposed by Braserski and Vaikuntanathan [3], which acts on blocks of plain-text rather than single bits.

## 4 Block-Wise Encryption

The main drawback in Gahí's method is that it requires an enormous number of homomorphic operations because it employs the DGHV encryption scheme, which uses bitwise encryption. We propose an alternative protocol called *Homomorphic Query Processing* that is compatible with the more recent ring-based fully homomorphic encryption scheme introduced by Braserski and Vaikuntanathan [3]. The major advantage is that Braserski's method works on plain-text and cipher-text blocks and thus the number of homomorphic operations required can be greatly reduced.

We first give a brief introduction to the ring based fully homomorphic Encryption Scheme proposed by Braserski, and then proceed to define our Homomorphic Query Processing method.

### 4.1 Ring Based Fully Homomorphic Encryption

This encryption scheme was introduced by Braserski and Vaikuntanathan [3] and operates on the polynomial ring  $R = \mathbb{Z}[X]/\langle f(x) \rangle$ ; the ring of polynomials with integer coefficients modulo  $f(x)$ , where,

$$f(x) = \prod_{\substack{1 \leq k \leq n \\ \gcd(k, n)=1}} \left( x - e^{2i\pi \frac{k}{n}} \right),$$

is the  $n^{\text{th}}$  cyclomatic polynomial. The plain-text space is the ring  $R_t = \mathbb{Z}_t[x]/\langle f(x) \rangle$ , where  $t$  is an integer. The key generation and encryption functions make use of two distributions  $\chi_{key}$  and  $\chi_{err}$  on  $R$  for generating small elements. The uniform distribution  $\chi_{key}$  is used in the key generation, and the discrete Gaussian distribution  $\chi_{err}$  is used to sample small noise polynomials. Specific details can be found in [3,8]. The scheme is based on the following algorithms.

- **KeyGen**( $n, q, t, \chi_{key}, \chi_{err}$ ): Operating on the input degree  $n$  and moduli  $q$  and  $t$ , this algorithm generates the public and private keys  $(pk, sk) = (h, f)$ , where  $f = [tf' + 1]_q$  and  $h = [tgf^{-1}]_q$ . Here, the key generation algorithm samples small polynomials from the key distribution  $f', g \rightarrow \chi_{key}$  such that  $f$  is invertible modulo  $q$  and  $[.]_q$  denotes coefficients of polynomials in  $R$  reduced by modulo  $q$ .
- **Encrypt**( $h, m$ ): Given a message  $m \in R$ , the Encrypt algorithm samples small error polynomials  $s, e \rightarrow \chi_{err}$  and outputs,  $c = [\lfloor q/t \rfloor [m]_t + e + hs]_q \in R$ , where  $\lfloor . \rfloor$  denotes the floor function.
- **Decrypt**( $f, c$ ): Given a cipher-text  $c$ , this algorithm outputs,  $m = \left[ \left[ \frac{t}{q} [fc]_q \right] \right]_t \in R$ .
- **Add**( $c_1, c_2$ ): Given two cipher-texts  $c_1$  and  $c_2$ , this algorithm outputs  $c_{add}(c_1, c_2) = [c_1 + c_2]_q$ .
- **Mult**( $c_1, c_2$ ): Multiplication of cipher-texts is performed in two steps. First, compute  $\tilde{c}_{mult} = \left[ \left[ \frac{t}{q} c_1 c_2 \right] \right]_q$ . However, this result cannot be decrypted to the

original plain-text using the decryption key  $f$ . Therefore, a process known as key switching is done to transform the cipher-text so that it can be decrypted with the original secret key. For more details, we refer to [8].

This encryption scheme is homomorphic with respect to addition and multiplication of plain-texts modulo  $t$ . The main advantage in using Braserski's encryption scheme is that it can be used to encrypt blocks of plain-text instead of dealing with single bits, as in the DGHV scheme [2]. For example, consider the block of plain-text bits, 10100. The integer representation of this block is the value 20. We can represent this integer using the polynomial  $X^2 + X^4 = \sum_{i=0}^4 2^i z_i$ , where  $z_i$  is the  $i^{\text{th}}$  bit of 10100. In general, if  $z$  is an integer and its binary representation is,  $z = (\pm 1) \sum_{i=0}^l 2^i z_i$ , where  $z_i \in \{0, 1\}$  and  $l = \lceil \log_2 |z| \rceil$ , then we can encode the number  $z$  as  $\sum_{i=0}^l z_i X^i \in R$ .

## 4.2 Converting the Plain-Text Space into a Field

As we shall see, in our *Homomorphic Query Processing* method, we invert certain plain-text elements and thus the plain-text space should be a field. Therefore, we now discuss how to convert the plain-text ring in Braserski's method to a field. Note that the plain-text space in Braserski's method is defined on the polynomial ring,  $R_t = \mathbb{Z}_t[x]/\langle f(x) \rangle$ . We shall select  $t = p$ , where  $p$  is a prime number. Then  $R_p$  is a field if and only if  $f$  is irreducible over  $\mathbb{Z}_p$ . Recall that  $f$  is the  $n^{\text{th}}$  cyclomatic polynomial defined as follows:

$$f(x) = \prod_{\substack{1 \leq k \leq n \\ \gcd(k, n) = 1}} \left( x - e^{2i\pi \frac{k}{n}} \right)$$

Let  $f(x) = (x - \alpha_1)(x - \alpha_2) \dots (x - \alpha_n)$  be a polynomial defined on  $\mathbb{Q}[x]$ . The discriminant of  $f$ , denoted by  $\Delta(f)$ , is defined [9] as,

$$\Delta(f) = \prod_{i < j} (\alpha_i - \alpha_j)^2$$

It has been proved in [9] that the  $n^{\text{th}}$  cyclotomic polynomial reduces modulo all primes if and only if the discriminant of the  $n^{\text{th}}$  cyclotomic polynomial is a square in  $\mathbb{Z}$ . Hence, by choosing a cyclotomic polynomial whose discriminant is not a square we can find a prime  $p$  such that  $f$  is irreducible over  $\mathbb{Z}_p$ . Furthermore, it is shown in [9] that whenever the discriminant of a cyclotomic polynomial  $f$  is not a square in  $\mathbb{Z}$ , there exist infinitely many primes such that  $f$  is irreducible over  $\mathbb{Z}_p$ . Thus, we can choose a cyclotomic polynomial with non-square discriminant and check for irreducibility using a standard polynomial irreducibility test such as Rabin's test, until we obtain a prime for which the cyclotomic polynomial is irreducible. For example, even if we consider a large cyclotomic polynomial with non-square discriminant like the  $107^{\text{th}}$  cyclotomic (which has degree 106), and consider the primes less than 100, it can be seen that it is irreducible over many primes:  $\mathbb{Z}_2, \mathbb{Z}_5, \mathbb{Z}_7, \mathbb{Z}_{17}, \mathbb{Z}_{31}, \mathbb{Z}_{43}, \mathbb{Z}_{59}, \mathbb{Z}_{67}, \mathbb{Z}_{71}, \mathbb{Z}_{73}$  and  $\mathbb{Z}_{97}$ .

We now propose our Homomorphic Query Processing scheme, which is compatible with the Braserski's ring based fully homomorphic encryption scheme mentioned previously.

## 5 Homomorphic Query Processing

We think of a database  $M$  as an  $\ell \times n$  matrix, where  $\ell$  denotes the number of records. We denote the record in the  $i$ -th row by  $M_i$ . Corresponding to each record  $M_i$  there are some attributes. One can think of the records as patients in a health database and the attributes are some test results. So,  $M_{i,j}$  denotes the result of test  $j$  for patient  $i$ . The database  $M$  is encrypted by the public key of database owner, Alice, who would like to search for a keyword  $m$ . The search will be over a specific attribute (column). Let us denote the values of this column by  $R_1, R_2, \dots$ . We write  $\bar{R}_i = \text{Enc}(R_i)$  for the  $\text{Enc}(R_i, pk)$ , where  $pk$  is the public key of database owner. Alice sends  $\bar{m}$  to the server and asks for the records such that  $\bar{m} = \bar{R}_i$ .

### 5.1 Unique Identifier

In this section, we consider a search on a column of the database where the entries of this column are all distinct. For example, we want the server to return a record with a specific ID.

Suppose that Alice wants to search for a message  $m$  over the database. First we assume that the query is contained somewhere in the database. The special scenario that the query is not found in the database is discussed in Sect. 5.2. For each  $i$ , the server computes the following:

$$F_i = \bar{1} \prod_{k \neq i} \frac{\bar{m} - \bar{R}_k}{\bar{R}_i - \bar{R}_k} \quad (5)$$

We remark that since the encryption is probabilistic, the chances that  $\bar{R}_i = \bar{R}_k$  is very slim and negligible. So the problem of dividing by zero is not an issue here. Since the  $R_i$ 's are all distinct, we have  $\bar{R}_i \neq \bar{R}_k$ , for all  $i, k$ . Note that here we are assuming that the query is contained somewhere in the database. We claim that either  $F_i = \text{Enc}(0)$  or  $F_i = \text{Enc}(1)$ . That is, whenever  $m = R_i$  (query being equal to the record we are comparing), we have  $F_i = \text{Enc}(1)$  and  $F_i = \text{Enc}(0)$ , otherwise. First we emphasize that due to the probabilistic property of encryption, we may not necessarily have  $\text{Enc}(R^{-1}) = (\text{Enc } R)^{-1}$ . Indeed, we have  $\text{Enc}(RR^{-1}) = \text{Enc}(1)$ . Thus,

$$\text{Enc}(R^{-1}) = \frac{\bar{1}}{\bar{R}},$$

that is  $\text{Enc}(R^{-1})$  can be expressed in this form as a fraction of encryption of 1 and encryption of  $R$ . To prove this claim, we note using the homomorphic property of encryption that

$$\begin{aligned}
F_i &= \bar{1} \prod_{k \neq i} \frac{\bar{m} - \bar{R}_k}{\bar{R}_i - \bar{R}_k} \\
&= \prod_{k \neq i} (\bar{m} - \bar{R}_k) \prod_{k \neq i} \frac{\bar{1}}{\bar{R}_i - \bar{R}_k} \\
&= \left( \prod_{k \neq i} \text{Enc}(m - R_k) \right) \left( \prod_{k \neq i} \text{Enc}(R_i - R_k)^{-1} \right) \\
&= \frac{\text{Enc} \prod_{k \neq i} (m - R_k)}{\text{Enc} \prod_{k \neq i} (R_i - R_k)}
\end{aligned}$$

The idea of defining the  $F_i$ 's in this way was inspired by Lagrange Interpolating Polynomials. If the query is not contained anywhere in the database, an encryption of something other than 1 or 0 will be the output. This special scenario is discussed in Sect. 5.2.

Now we consider the sequence

$$R' = \sum_i \bar{R}_i F_i,$$

which give us the only encrypted record that matches our search query.

## 5.2 Query Not Found in the Database

As promised previously, we now look at the special case where the record that is searched for is not contained anywhere in the database. In this case the value  $F_i$  will be something other than an encryption of 1 or 0. These garbage encrypted values will carry themselves into the rest of the protocol, resulting in Eq. (7) with a nonsensical sequence. Hence, if Alice receives a nonsensical sequence as the final result, it implies that the record that was searched is not contained in the database. As an alternative approach, we can compute  $\prod_i (\text{Enc}(m) - \text{Enc}(R_i))$  prior to computing the  $F_i$  in Eq. (5) and send it to Alice to decrypt. If the result is zero then  $m$  is contained in the database, and if it is non-zero,  $m$  is not contained in the database and therefore Alice can send a message to the server to abort the search.

## 5.3 General Case

Given a message  $m$ , suppose now that Alice wants to search for all  $\bar{R}_i$  such that  $R_i = m$ . We may attempt to adopt the same method as in Sect. 5.1, however we encounter a problem computing the  $F_i$  in Eq. (5). Indeed, in the case we have multiple matches or the case where  $R_k = R_i$ , we run into a problem.

Recently Kim et al. [10] considered equality testing on the cipher texts. Suppose now we have a function EQUAL that as input takes two cipher texts  $c_1$  and  $c_2$ , where  $c_1 = \text{Enc}(m_1)$  and  $c_2 = \text{Enc}(m_2)$ . The output of EQUAL is  $\text{Enc}(1)$  if  $m_1 = m_2$  and  $\text{Enc}(0)$ , otherwise.

Now we want to compare and see whether  $m$  is equal to each  $R_k$ . So we compute  $F_k = \text{EQUAL}(\bar{R}_k, \bar{m})$ , for each record  $R_k$ . We can now return to user the vector that in its  $k$ -th position has  $\bar{R}_k F_k$ . This vector after decryption yields a long vector consisting mostly of zeros except when  $R_k$  is equal to  $m$  in which case the  $k$ -th position of the decrypted vector is  $R_k$ . This is not a plausible way to return the search outcome. Ideally we want to return only the records that match  $m$ . To accomplish this task, we need to do some further computations. First, for each positive integer  $k$ , we calculate

$$G_i = \sum_{j \leq i} F_j.$$

So,  $G_i$  (in the encrypted form) indicates how many times a match to  $m$  is found so far. In other words, if we are comparing  $R_k$  and  $m$  and so far  $r$  records match  $m$ , then  $G_i = \text{Enc}(r)$ .

Next, for each record  $R_k$ , we construct a vector  $E_k$  such that  $E_k$  in the  $i$ -th position has  $F_k \text{EQUAL}(G_k, \bar{i})$  and  $\text{Enc}(0)$  elsewhere. Then we form  $R' = \sum_k \bar{R}_k E_k$ .

An alternative way of constructing  $R'$  without using the EQUAL function is as follows. We define the partial sums of the  $F_i$  values as follows:

$$G_i = \sum_{j \leq i} F_j. \quad (6)$$

Using these partial sums, we can then calculate the sequence  $F'_i = (F'_{i,k})$  corresponding to each record as follows,

$$\begin{aligned} F'_{i,k} &= F_i \left( \prod_{j \neq k} G_i - \text{Enc}(j) \right) \left( \text{Enc} \prod_{j \neq k} (k-j)^{-1} \right) \\ &= F_i \left( \prod_{j \neq k} G_i - \bar{j} \right) \prod_{j \neq k} \frac{1}{\bar{k} - \bar{j}} \\ &= \bar{1} F_i \prod_{j \neq k} \frac{G_i - \bar{j}}{\bar{k} - \bar{j}}, \end{aligned}$$

where  $1 \leq k \leq i$ . It can be seen that  $F'_{i,k} = \text{Enc}(1)$  if  $F_i = \text{Enc}(1)$  and  $G_i = \text{Enc}(k)$  are both satisfied. Hence, the sequence  $F'_i$  has the property that whenever  $F_i = \text{Enc}(1)$  (i.e., the  $i^{\text{th}}$  record matches the query), we have an  $\text{Enc}(1)$  at the  $k^{\text{th}}$  position of the sequence where  $G_i = \text{Enc}(k)$ . All other entries of the sequence are encryptions of zero. Finally we consider the sequence

$$R' = \sum_i \bar{R}_i F'_i, \quad (7)$$

where  $R_k$  is the  $k$ -th record in  $D$  will give us a sequence containing only the encrypted records that match our search query. Note that the definition of  $R'$

relies on adding vectors of different lengths. This is done in the natural way, whereby all the vectors are made the same length by padding with zeros prior to addition.

To further illustrate our scheme, let us consider an example where the database contains five records, each with 4 bits of data. Also, let our encryption scheme encrypt 2 bits at a time. Then, if the search query is  $(\text{Enc}(2), \text{Enc}(3))$ , the corresponding  $F_i$  and  $G_i$  values are given in Table 2.

**Table 2.** Sample database and corresponding  $F_i$  and  $G_i$  values

Database records	$F_i$	$G_i$
$(0, 0, 1, 0)$	$\text{Enc}(0)$	$\text{Enc}(0)$
$(1, 0, 1, 1)$	$\text{Enc}(1)$	$\text{Enc}(1)$
$(1, 0, 0, 1)$	$\text{Enc}(0)$	$\text{Enc}(1)$
$(1, 0, 1, 1)$	$\text{Enc}(1)$	$\text{Enc}(2)$
$(1, 1, 0, 0)$	$\text{Enc}(0)$	$\text{Enc}(2)$

The resulting sequences  $(F'_i)$  would be similar as in Gahi's scheme,

$$F'_1 = (\text{Enc}(0))$$

$$F'_2 = (\text{Enc}(1), \text{Enc}(0))$$

$$F'_3 = (\text{Enc}(0), \text{Enc}(0), \text{Enc}(0))$$

$$F'_4 = (\text{Enc}(0), \text{Enc}(1), \text{Enc}(0), \text{Enc}(0))$$

$$F'_5 = (\text{Enc}(0), \text{Enc}(0), \text{Enc}(0), \text{Enc}(0), \text{Enc}(0)).$$

Therefore, the sequence  $R'$  would be,

$$R' = (\text{Enc}(R_2), \text{Enc}(R_3), \text{Enc}(0), \text{Enc}(0), \text{Enc}(0))$$

At this point, the sequence  $R'$  will contain all the records that match our query but with trailing encryptions of zeros which we do not need. Hence, a second sum is calculated at the server side to determine the number of terms that are useful in the sequence:

$$\text{Enc}(n) = \sum_r F_r.$$

Then  $\text{Enc}(n)$  will be returned to Alice and decrypted to obtain the number of records that match the search query. Hence, the sequence  $R'$  can be truncated at the appropriate point and returned to Alice for decryption.

It should be noted that the server will know the number of records that match Alice's query. We believe that this information is not sufficient for the server to gain any additional information about the search query. Alternatively, we could return the whole sequence without truncation, keeping the number of matching records private from the server. However, the communication overhead will be increased significantly in this case, since the length of the sequence will be equal to the number of records in the database.

## 6 Complexity

Our scheme has the main advantage of having the potential to be used with more recent fully homomorphic encryption schemes rather than being restricted to the DGHV scheme. This gives the flexibility to use our method with block based encryption schemes such as Brakerski's [3], which reduces the number of encryption steps. For example, referring back to Eq. (1), we can see that the  $I_t$  values are calculated by comparing the query with each record bit-wise. If there are  $\ell$  records in the database and each of them are encrypted using  $r$  bits, the number of operations that are required to calculate all the  $I_t$  values will be  $\mathcal{O}(r\ell)$ . In our method, Eq. (5) acts as the analogue of Eq. (1). However, the encryptions are done block-wise in our scheme, and hence the number of operations it would take to calculate the  $F_i$  value in Eq. (5) will be  $\mathcal{O}(\ell)$ . For Eq. (2) in Gahis method, the number of operations that should be performed to calculate all the partial sums will be  $\mathcal{O}(r\ell^2)$ , since there are  $\mathcal{O}(\ell^2)$  multiplications and each multiplication should be done bit-wise; whereas the calculation of partial sums in our scheme (Eq. (6)), the number of operations is reduced to  $\mathcal{O}(\ell^2)$ . Thus, it can be seen that in each step of our scheme, the number of operations performed is reduced by a factor of  $r$  compared to Gahis method.

## References

1. Gah, Y., Guennoun, M., El-Khatib, K.: A secure database system using homomorphic encryption schemes. In: The Third International Conference on Advances in Databases, Knowledge, and Data Applications (2011)
2. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-13190-5\\_2](https://doi.org/10.1007/978-3-642-13190-5_2)
3. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-22792-9\\_29](https://doi.org/10.1007/978-3-642-22792-9_29)
4. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**, 120–126 (1978)
5. Hesse, H., Matthies, C.: Introduction to homomorphic encryption. In: Cloud Security Mechanisms, December 2013
6. Gentry, C.: A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University (2009)
7. Boneh, D., Gentry, C., Halevi, S., Wang, F., Wu, D.J.: Private database queries using somewhat homomorphic encryption. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) ACNS 2013. LNCS, vol. 7954, pp. 102–118. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-38980-1\\_7](https://doi.org/10.1007/978-3-642-38980-1_7)

8. Bos, J.W., Lauter, K., Loftus, J., Naehrig, M.: Improved security for a ring-based fully homomorphic encryption scheme. In: Stam, M. (ed.) IMACC 2013. LNCS, vol. 8308, pp. 45–64. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-45239-0\\_4](https://doi.org/10.1007/978-3-642-45239-0_4)
9. Harrison, B.: On the reducibility of cyclotomic polynomials over finite fields. *Am. Math. Mon.* **114**, 813–818 (2007)
10. Kim, M., Lee, H.T., Ling, S., Wang, H.: On the efficiency of FHE-based private queries. *IEEE Trans. Dependable Secure Comput.* **15**(2), 357–363 (2018)



# A Cache-Aware Congestion Control for Reliable Transport in Wireless Sensor Networks

Melchizedek I. Alipio<sup>(✉)</sup> and Nestor Michael C. Tiglao

Ubiquitous Computing Laboratory, EEE Institute,  
University of the Philippines Diliman, 1101 Quezon City, Philippines  
`{melchizedek.alipio,nestor}@eee.upd.edu.ph`

**Abstract.** Data caching and congestion control are two strategies that can enhance the transport reliability in constrained Wireless Sensor Networks. However, these two mechanisms are designed independently for most transport protocols developed for WSN. This work developed a new cache-aware congestion control mechanism for reliable transport. RT-CaCC utilizes cache management policies such as cache insertion, cache elimination and cache size to mitigate packet losses in the network while maximizing cache utilization and resource allocation. It uses two cache management policies for packet loss detection: implicit notifications and expiration of timeout. In addition, it utilizes congestion avoidance using cache-aware rate control mechanism employing transmission window limit as a function of cache size. Results showed that the RT-CaCC obtained significant improvement gain in terms of cache utilization, end-to-end delay and throughput performance specifically during high level of packet loss in the network.

**Keywords:** Cache-aware • Congestion control • Intermediate caching  
Internet of Things • Wireless Sensor Networks

## 1 Introduction

A typical Wireless Sensor Network (WSN) consists of tiny nodes that are equipped with embedded computing devices interfacing with sensors or actuators. These sensor networks are vital component of Internet of Things (IoT) and characterized as constrained networks due to limited memory, computing and energy capability. A sizable set of these nodes is dispersed over a wide geographical area to monitor a physical or environmental event. Therefore, packets generated at source nodes are usually transmitted to the sink through multi-hop communication [1]. In effect, these networks experience high probability of packet drop due to poor link quality, link contentions or buffer overflows. Thus, an effective transport protocol is a must. One mechanism of improving reliability is intermediate caching thru local retransmissions. Another mechanism is to utilize congestion control strategies which can alleviate packet losses due to

network congestion. Some transport protocols combined these two mechanism to further enhance the reliability of data transport in WSN.

To the best of our knowledge, this paper makes the following contributions: (1) design a cache-aware congestion control mechanism based on different cache management policies which are utilized to optimize cache utilization of the transport protocol. To achieve this intention, we (2) perform simulations to evaluate and analyze the performance gain improvement of the congestion control mechanism as compared with the baseline protocols.

The rest of the paper is organized as follows: Sect. 2 discusses some of the related works. In Sect. 3, we discuss transport layer caching and the cache-aware congestion control. In Sect. 5, we discuss the simulation environment, results and discussions. Finally, Sect. 6 concludes the paper.

## 2 Related Work

Existing transport protocols in WSN combined both reliability and congestion control mechanisms to improve the performance of packet transmissions. Increasing the reliability is achieved by using local retransmission at intermediate nodes. While congestion control alleviates the network during high level of packet losses due to poor wireless quality, contending flows or buffer overflow. DTC [2] and TSS [3] modified TCP protocol in order to provide direct TCP/IP - WSN compatibility which is implemented in the intermediate nodes and requires no protocol changing in the end nodes. It uses intermediate caching and provides hop-by-hop reliability. Segments are cached at the intermediate nodes based on the highest sequence number and link layer feedback. However, it uses AIMD rate control mechanism to mitigate packet losses during congestion states. Therefore, it does not classify packet losses due to wireless link error which makes the congestion window oscillates aggressively.

On the other hand, ERCTP [4] implements a modular approach for congestion control and reliability mechanisms. Its source rate adjustment module defines the new transmission rate adjustment for child nodes in order to mitigate congestion. It monitors the instantaneous network statistics which helps sink to explicitly and periodically send the estimated value of rate adjustment to source nodes, which is obtained based on congestion index calculation. The use of explicit notification to infer possible congestion may lead to additional traffic load to the network. Thus affecting the throughput performance of the transport protocol. Moreover, RCRT [5] also uses AIMD while it adapts the total aggregate rate of all the flows as observed by the sink, rather than the rate of a single flow. Whenever RCRT determines the network is congested, it applies the rate decrease step by time-dependent multiplicative decrease factor, computes a new rate allocation for all the flows, and sends the new rate for each flow to the corresponding source.

Intermediate caching and congestion control mechanisms are different techniques that effectively improve reliability of data transport performance in WSN. However, these two mechanisms are designed independently for most transport

protocols developed. In previous works, congestion control techniques are not cache-aware. This can possibly result to non-optimal use of intermediate caching, inappropriate congestion window movement and increase in energy consumption of intermediate nodes. To the best of the our knowledge, no study has yet to develop a cache-based transport protocol that has an appropriate congestion control mechanism that can improve cache utilization, network efficiency, and resource allocation.

### 3 Overview of Transport Data Caching and Management Policies

Intermediate caching alone can mitigate packet losses either due to contentions or congestion from local retransmissions up to a certain optimal transmission window size [6]. In addition, the information from cache elimination policies and cache size can be used to adapt the source rate control in improving the performance of a cache-based transport protocol in terms of throughput, end-to-end delay and cache utilization [7]. These ideas provide the underlying support for the new cache-aware congestion control integrated in a transport protocol (RT-CaCC).

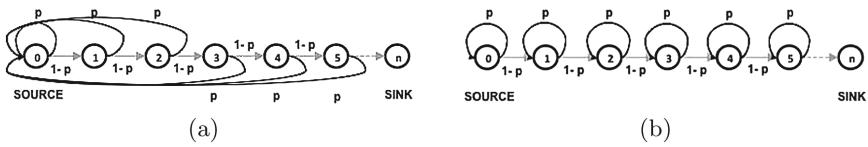
The caching mechanism of the RT-CaCC protocol was inspired by the DTSN<sup>+</sup> [8]. RT-CaCC uses both positive acknowledgment (ACK) and selective negative acknowledgment (NACK) to be sent from the receiver upon the request of the sender through an Explicit Acknowledgment Request (EAR). The EAR signal is piggybacked on to a data packet. After sending an EAR, the source launches an EAR timer. If the EAR timer expires before an ACK/NACK is received, the source retransmits the EAR packet. Upon the reception of EAR at the receiver node, a NACK, which contains a bitmap of missing packets, is generated and transmitted back to the sender. While relaying such NACKs, intermediate nodes learn about the missing packets and check if those packets are present in their cache. If so, the intermediate nodes retransmit those packets towards the receiver and modify the NACK bitmap accordingly before sending it towards the sender. Likewise, RT-CaCC adapts a NACK repair mechanism whereby intermediate nodes can issue NACK signals to hasten the repair process. In addition to receivers being able to detect lost packet, intermediate nodes detect packet loss and signal the previous hop node through a repair negative acknowledgment (RNACK) control packet that contains the sequence number of lost packet. Upon receiving the RNACK, the previous hop node will retransmit the lost packet towards the destination if a copy is found in the cache. If not, the RNACK will be propagated towards the source. The sending of the RNACK is not timer-driven but triggered as soon as an out-of-sequence packet is detected. This feature further reduces the probability of packet loss from poor wireless link errors.

At intermediate nodes, RT-CaCC used cache insertion policy with a certain probability which can be based on cache partitioning scheme [9]. The probability value must be chosen to maximize cache utilization of data packets requested by

the NACK along the reverse path. At each intermediate node, the total cache size  $CS$  is divided among the different flows that cross the node. Let  $\omega_i^n \geq 0$  be a weight related to the fraction of cache at node  $n$  that is assigned to flow  $i$ . The actual fraction is given by the normalized weight  $\rho_i^n = \frac{\omega_i^n}{\sum_{j=1}^{F_n} \omega_j^n}$ . A packet that belongs to flow  $i$  can only be cached in the fraction of cache assigned to flow  $i$ , whose size is equal to  $\rho_i^n \times CS$ . Consequently, the caching probability  $P_{cache}$  for a packet that belongs to flow  $i$  at node  $n$  is given by  $\min(1, \rho_i^n \times CS)$ . Considering that each node in the network is simultaneously crossed by more than one flow wherein all flows have equal hop length, an equitable way for partitioning the cache is to divide it equally between the flows. In this case, a uniform cache partitioning can be implemented where each flow is assigned the same  $\omega_j^n$  to achieve fairness. However, for complex scenarios with higher number of concurrent flows, heterogeneous link quality and length, a non-uniform cache partitioning can be used [9] and is out of the scope of this study.

RT-CACC also used cache elimination policy in the form of implicit notifications that the sender nodes receive: ACK ( $holes_{ACK} = 0$ ), NACK without holes ( $holes_{NACK} = 0$ ) and NACK with holes ( $holes_{NACK} \neq 0$ ). Holes represent the number of packets that were not successfully retransmitted by intermediate caching. Therefore, the list of packets in the hole is retransmitted by the sender together with the new batch of packets. From the three notifications, the reception of NACK with holes indicates a high degree of packet loss that the local retransmissions from intermediate caching was not able to handle.

The probability of success in an end-to-end transmission can be evaluated by the number of hops  $H$  and probability of packet loss  $P_{loss}$  as  $(1 - P_{loss})^H$  shown in Fig. 1a [10]. When caching is used at intermediate nodes and assuming that the given  $CS$  can store all the packets needed for local retransmission, the expected number of transmissions (ENT) as shown in Fig. 1b is given by  $H \cdot \sum_{i=0}^{\infty} P_{loss}^i$ .



**Fig. 1.** Data packet transmissions probabilistic model: (a) without intermediate caching and (b) with intermediate caching

However, since intermediate nodes are constrained and  $CS$  has limited capacity, the number of packets to be inserted into the cache can be based on a certain probability. This caching probability can be based on cache partitioning scheme  $P_{cache}$ . Therefore, the actual expected number of transmissions is given by

$$ENT_{cache} = H + \sum_{i=1}^{\infty} P_{loss}^i \cdot \sum_{h=1}^H \left[ 1 + P_{cache}^{h-1} \cdot ENT(h-1) + \sum_{j=2}^{h-1} P_{cache}^{h-j} \cdot ENT(h-j) \right] \quad (1)$$

where  $ENT$  is the expected number of transmission without caching which is given by

$$ENT = H + \left[ \frac{\sum_{h=1}^{H-1} P_{loss} \cdot (1 - P_{loss} \cdot h)}{1 - (1 - P_{loss})^H} + 1 \right] \cdot \sum_{i=1}^{\infty} (1 - (1 - P_{loss})^H)^i \quad (2)$$

In this case, retransmission can be performed from any of the nodes in the forward path that are behind the hop link where the packet loss occurred taking into consideration the best possible value in  $P_{cache}$ .

## 4 Cache-Aware Congestion Control

The context of cache-aware is based on different cache management policies which are utilized by the congestion control mechanism to mitigate packet losses while optimizing cache utilization and bandwidth allocation. This approach assumes that the network is uncongested as long as end-to-end losses from transient congestion and poor wireless links are repaired immediately. Furthermore, it permits the source to transmit at a higher rate even if there are occasional end-to-end losses, since these losses can be recovered by intermediate caching.

The RT-CaCC protocol utilizes cache-aware strategies to detect, notify and avoid packet losses due to poor wireless channel or buffer congestion in the network discussed as follows:

### 4.1 Packet Loss Detection Using Cache Elimination Policy

RT-CaCC utilizes implicit notifications in the form of cache elimination policy to infer the approximate degree of packet losses. NACK notifications can provide the most updated packet loss level in the network as they know the number of packets already recovered through the updated bitmap as they travel along the return path and the number of packets (holes) which are not. This strategy eliminates the need for channel probing while making the most effective use of cache space. Therefore, it will lessen the sensor node's memory and computing requirements, energy consumption and maximize cache utilization. However, in this strategy, there is no way to differentiate the packet losses. In low power and lossy types of networks such as WSN, previous results show that when congestion occurs, the majority of packets are lost due to node level congestion as compared to link level contention [6]. Therefore, RT-CaCC uses a second strategy to mitigate packet losses mainly due to buffer congestion.

### 4.2 Packet Loss Detection Using EAR Timeout ( $ETO$ )

The expiration of EAR timer is used to detect node-level congestion which predominates during high level of network traffic in WSN. EAR timeout  $ETO$  is dynamically set using congestion round-trip time (RTT). In ad hoc wireless networks, RTT is composed of processing, queuing, transmission, propagation and

contention delays. To estimate the congestion RTT of a segment, RT-CaCC marks the time when the frame reaches the output queue header at the intermediate node. The estimation only uses the output queue since it is assumed that the input queue only refers to the routing path availability and discovery and is out of the scope of this work. RT-CaCC does not include contention delay in the RTT estimation to ensure that the delay is mainly contributed by buffer overflow. The total time from propagation to transmission delay is designated as one-hop delay of a packet at node  $n$  indicated as  $d_h^n$  assuming the clock of adjacent nodes is synchronous. For  $N$  number of hops in the forward path, the previous  $d_h^n$  is added to the current  $d_h^n$  until it reaches the sink node. Once the sink node received the EAR notification piggybacked in the last packet, the sum of all  $d_h^n$  for  $M$  number of packets in a segment is computed as  $T_{delay-DATA}$ . Afterwhich, an implicit notification either ACK or NACK is propagated in the return path until it reaches the destination and the total time of propagation corresponds to  $T_{delay-ACK}$ . Therefore, the total congestion RTT is given by  $RTT = T_{delay-DATA} + T_{delay-ACK}$ .

When the source node transmits the EAR notification, an EAR timeout is set automatically. In order to dynamically set the EAR timeout, total congestion RTT is used taking into consideration the additional delay caused by local retransmissions from intermediate caching. In WSN, the local retransmissions performed by intermediate caching at wireless link can cause the end-to-end RTT to increase significantly in a short time due to the long processing delay. In effect, huge time gaps in receiving ACK and NACK at the sender nodes is experienced. When there are no packet loss and an  $holes_{ACK}$  is received at the sender, the estimated delay variation is minimal and EAR timer is set accordingly. On the other hand, if there are frequent packet loss occur, a  $holes_{NACK} \neq 0$  is received at the sender and local retransmissions are triggered, the estimated EAR timer should be adjusted in order to handle delay variation caused by local retransmissions from intermediate caching. Therefore, an additional delay interval  $\delta_{delay}$  is computed as a function of estimated congestion RTT, probability of packet loss and caching probability in (1) leading to  $\delta_{delay} = RTT \times ENT_{cache}$ . It can be noticed that when  $P_{loss}$  and  $P_{cache}$  approaches to 1, most packets are being lost and are need to be recovered. Therefore, end-to-end delay should be increased by at least another RTT in order for these lost packets to be recovered. To attain this,  $ETO$  is computed at the sender as

$$ETO = RTT + \delta_{delay} \quad (3)$$

The expiration of  $ETO$  before an ACK or NACK is received at the sender node infers a buffer congestion. The sender node will act by retransmitting an EAR packet towards the sink node and reduces its transmission rate accordingly. This approach could be seen as an extension of the classical TCP algorithm. However, instead of constants that are used to take into account history of the current state, dynamically changing parameter is used.

### 4.3 Congestion Avoidance at the Source Node

At the sender node, a cache-aware rate control strategy was used by RT-CaCC based on packet loss detections described in the previous sections. It also used a bounded congestion window size based on bandwidth-delay product (BDP) and  $CS$ . The idea is to limit the upper transmission window size to the average BDP of the network which serves as the congestion window limit ( $CWL$ ). BDP is an important indicator of network capacity which refers to the maximum number of bits a connection can accommodate. Therefore, the total number of outstanding data packets, like in-flight or unacknowledged ones, cannot exceed this upper bound. For constrained networks like WSNs, it is important that a transmission window limit must be used and tuned to an appropriate value to ensure sufficient pipelining and to avoid the risk of overloading the network. In addition, RT-CaCC used  $CS$  value as the minimum transmission window size to optimize cache utilization since moving the window below  $CS$  can lead to sub-optimal cache performance. Assuming that  $CS$  is always less than the total buffer size  $B$ , cache-based transport protocol starts to obtain optimum cache utilization when the transmission window is equal to  $CS$ . On the other hand, RT-CaCC can achieve the optimum throughput at  $CWL$  as a function of  $B$ .

RT-CaCC used the average BDP ( $BDP_{ave}$ ) as the upper bound of  $CWL$  as a function of congestion RTT discussed previously. RT-CaCC only considers BDP determined by the time that data packets flow continuously. In this case, contention delay is not included since it is the period where packets are temporarily blocked by the node when contending to access the wireless channel to send the data and is not an indicator of available bandwidth capacity. Therefore, RT-CaCC only uses the output buffer in the determination of  $BDP_{ave}$ . However, to ensure optimum cache utilization,  $BDP_{ave}$  is only used if  $BDP_{ave} > W_{CS}$  wherein  $W_{CS}$  is the window size equal to  $CS$ . Since packets spend longer time passing through a bottleneck link, RT-CaCC measures the available bandwidth based on its share at this link. Since the BDP carried out by each packet is continuously changing, RT-CaCC computes the  $BDP_{ave}$  at the sender node using  $BDP_{ave} = \frac{\sum(\frac{S}{d_{max}}) \times RTT}{M}$  where  $S$  is the packet size,  $M$  is the total number of observed packets and  $d_{max}$  is  $\max(d_h^n), 1 \leq n \leq N$ . In order to achieve this, RT-CaCC protocol used additional packet header fields.

Let  $R(t)$  denote the rate allocated for current congestion window ( $W_t$ ) which is calculated at the sender node once implicit notifications are received. The rate control mechanism used an AIMD on  $R(t)$  which is counteractive with the traditional TCP-AIMD. When the network experience packet losses either due to poor channel quality or link contentions, intermediate caching will act primarily to perform local retransmissions. The rate control mechanism of RT-CaCC protocol is summarized in Algorithm 1.

If the sender node receives an ACK or NACK without holes,  $R(t)$  additively increases as a function of  $\alpha$ . If the source node receives NACK with holes,  $R(t)$  multiplicatively decreases by  $\beta$ . The idea is to dynamically adjust the transmission rate according to packet loss level. This will continue until  $W_t$  reaches  $BDP_{ave}$ . On the other hand, the expiration of  $ETO$  will decrease  $W_t$  equal  $W_{CS}$ .

**Algorithm 1.** Rate Control Algorithm at the Source Node

---

```

1: procedure PKT_RECV(PKT)
2:   %compute the value of RTT and  $\delta_{delay}$ 
3:   %compute the value of ETO from previous segment
4:   %set ETO after transmission of last packet
5:   if (ETO expires) then
6:     decrease current window to minimum limit  $W_{CS}$ 
7:     %retransmit EAR notification to sink node
8:   else
9:     if (packet header received is  $holes_{NACK} \neq 0$ ) then
10:       if (check current window  $W_t > W_{CS}$ ) then
11:          $R(t) = [(W_t - W_{CS}) \times \beta] + W_{CS}$ 
12:       else
13:         set current window to minimum limit  $W_{CS}$ 
14:       end if
15:     else
16:       %compute the value of  $BDP_{ave}$ 
17:       %use  $BDP_{ave}$  if  $BDP_{ave} > CS$ ; otherwise  $CWL = CS$ 
18:       if (check current window  $W_t < BDP_{ave}$ ) then
19:          $R(t) = W_t + (\alpha \times \frac{S}{RTT})$ 
20:       else
21:         set current window to maximum limit  $BDP_{ave}$ 
22:       end if
23:     end if
24:   end if
25: end procedure

```

---

The idea is to move  $W$  between  $BDP_{ave}$  and  $W_{CS}$  to prevent network overload while optimizing cache utilization. The values of  $\alpha$  and  $\beta$  use increase-by-one and decrease-to-half strategy, respectively, to lessen the effect of aggressive window oscillation that can lead to low network resource utilization.

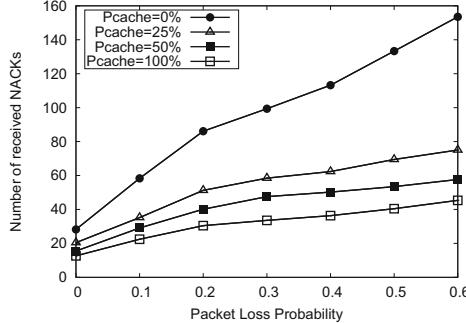
## 5 Simulations and Results

Simulations in NS-2 were carried out to evaluate the performance of the RT-CaCC protocol. For the network scenario, it is assumed that all intermediate nodes have  $CS$  equal to 10 packets while buffer size is set to 50 packets (default in NS-2). Packet size is set to 500 bytes and each topology used fixed routing (FIXRT) with Binary Symmetric Channel (BSC). Since 802.11b set to 1 Mbps MAC protocol was used, RTS/CTS are disabled and the number of retries was set to 4 in order to make it as similar as possible to 802.15.4 MAC protocol standard for WSN. To evaluate the performance of the RT-CaCC, end-to-end delay, throughput, transmission cost and cache hit are used as primary metrics. *Cache hit* refers to the number of times the data are requested from the intermediate nodes cache memory while *transmission cost* is the total number of data packets, transport layer ACK and NACK packets and MAC layer ACKs per total number of packets that need to be delivered end-to-end.

### 5.1 Varying $P_{cache}$

The number of received NACK at the sender node was evaluated while varying the caching probability in a 10-node linear topology. From Fig. 2, for 100%  $P_{cache}$ ,

fewer number of retransmissions is required than with  $0\%P_{cache}$ . It can be observed that using 50% probability is still efficient in reducing the number of packet retransmissions. However, lower  $P_{cache}$  can lead to more retransmissions thus requiring more energy consumption from the congestion control. Appropriate value of  $P_{cache}$  should be taken into account due to its effect in RTT and *ETO* variations. Larger  $P_{cache}$  supports more local retransmission from intermediate. Thus, it requires longer RTT and *ETO* expiration time.



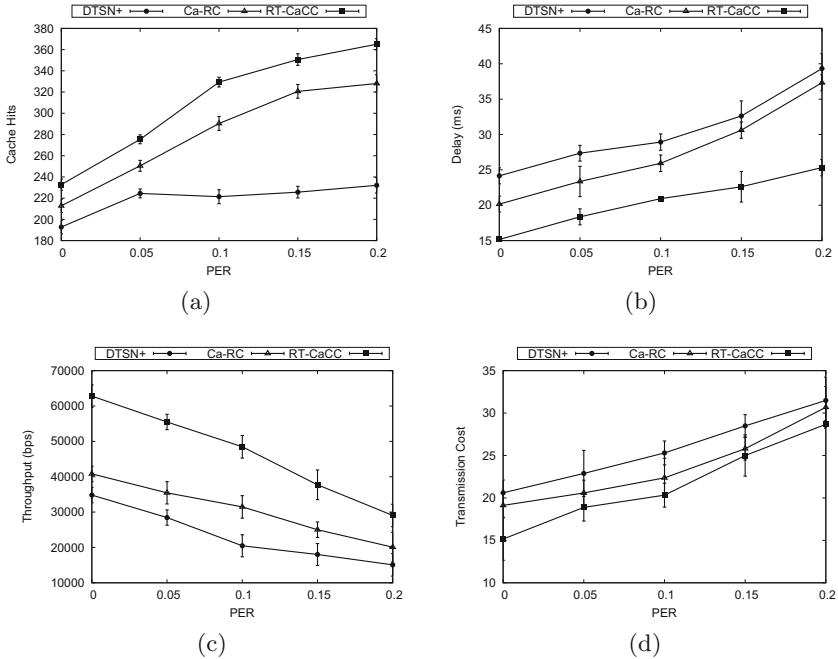
**Fig. 2.** Number of received NACKs at the sender node

## 5.2 Varying Packet Error Rates

To evaluate the improvement gain of the congestion control mechanism, the RT-CaCC was simulated against the original DTSN<sup>+</sup> with fixed transmission window and the modified DTSN<sup>+</sup> with dynamic cache-aware rate control mechanism designated as Ca-RC [7]. 50% $P_{cache}$  is uniformly allocated to all intermediate nodes. A linear topology composed of 10 nodes wherein 8 nodes served as intermediate caching nodes with 20% packet error rate was used. RT-CaCC gained 48.09% and 30.88% throughput improvement gain against DTSN<sup>+</sup> and Ca-RC, respectively. It also achieved 35.61% and 26.15% end-to-end delay improvement gain as compared with DTSN<sup>+</sup> and Ca-RC, respectively. Finally, 36.43% and 10.13% cache hits improvement gain were obtained better than DTSN<sup>+</sup> and Ca-RC, respectively. The addition of *ETO* component of RT-CaCC significantly maximized cache utilization by adapting to variations in RTT due to local retransmissions. Although the minimal throughput gain can be attributed to conservative congestion window, this can be further improved by setting the appropriate  $P_{cache}$  (Fig. 3).

## 5.3 Link Contentions with Multiple Flows

Figure 4 shows the results of varying the number of flows in contending flow topology. In terms of delay, the cache-aware congestion control performed well with 34.76% and 27.97% average gain difference than DTSN<sup>+</sup> and Ca-RC,

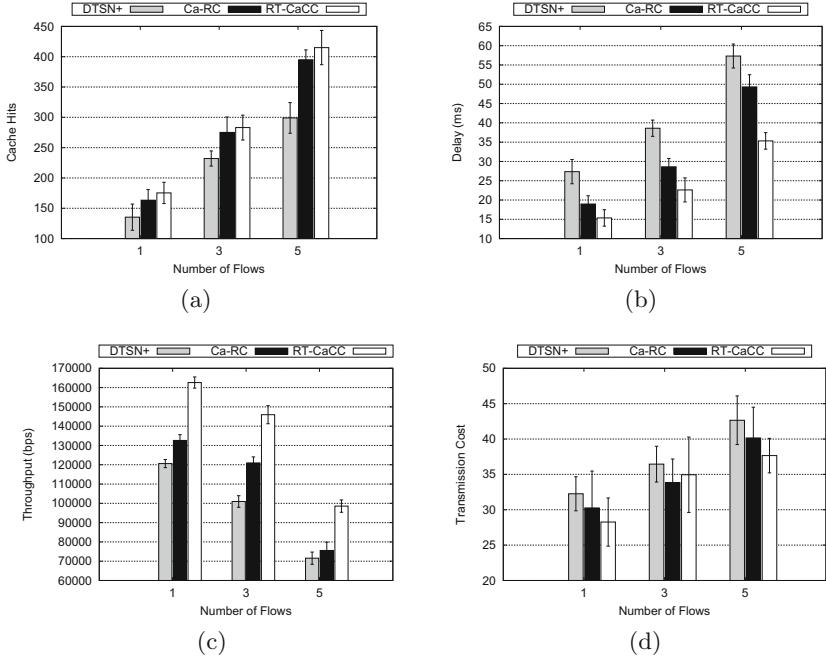


**Fig. 3.** RT-CaCC performance with varying PER: (a) cache hits, (b) end-to-end delay, (c) throughput and (d) transmission cost

respectively, at worst case. It shows that the cache-aware retransmission timeout mechanism of RT-CaCC is effective enough to reduce the end-to-end delay which is very important in an event-based applications. In addition, RT-CaCC was able to prevent the network from collapsing and recover immediately specifically at higher buffer overflows. It is mainly evident from the high throughput performance gain obtaining 20.92% and 10.13% as compared with DTSN<sup>+</sup> and Ca-RC, respectively. In terms of cache hits, RT-CaCC achieved 18.71% and 9.97% gain improvement against DTSN<sup>+</sup> and Ca-RC, respectively. It can be deduced that using cache elimination policy such as implicit NACK notification is effective in mitigating link-level congestion from contending flows than its predecessor protocols.

#### 5.4 Bottleneck Link with Multiple Flows

Two RT-CaCC flows were also evaluated in a bottleneck topology entering the link while varying cache size values. Figure 5 shows the results wherein 20% and 10% cache hit improvement gain were achieved by the RT-CaCC as compared with DTSN<sup>+</sup> and Ca-RC, respectively. The protocol also gained 47% and 25% end-to-end delay improvement gain against DTSN<sup>+</sup> and Ca-RC, respectively. Further, it also achieved 14% and 5% throughput improvement gain compared

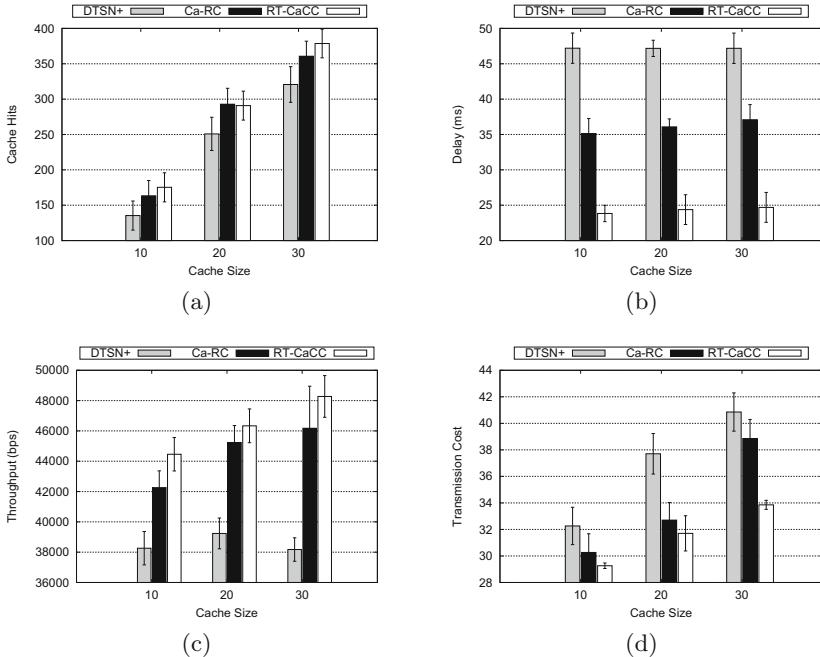


**Fig. 4.** RT-CaCC performance with varying number of flows: (a) cache hits, (b) end-to-end delay, (c) throughput and (d) transmission cost

to DTSN<sup>+</sup> and Ca-RC, respectively. This only shows that the *ETO* mechanism which uses congestion RTT is effective enough to detect and mitigate congestion due to buffer overflow. It can also be deduced that the value of *CS* affects the behavior of RT-CaCC protocol. Ideally, larger *CS* should be used since it will also increase the number of retransmissions for successful recovery of packet losses. However, in real-life scenario, smaller *CS* should be used due to the constrained characteristics of WSN.

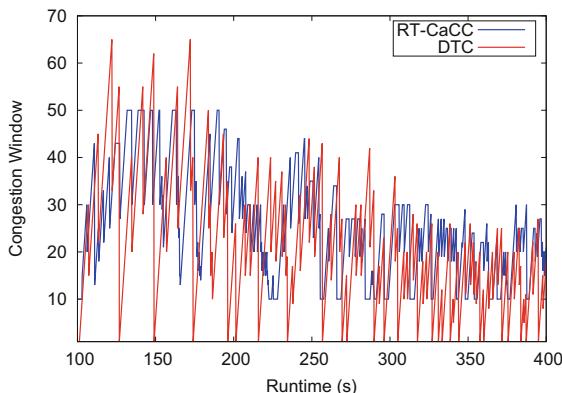
## 5.5 Congestion Window Response

The congestion window behavior of RT-CaCC was also observed in a bottleneck link topology with increasing number of flows. Figure 6 shows the congestion window response of RT-CaCC and DTC protocols. DTC [2] was a modified TCP with caching mechanism which uses the traditional TCP-AIMD algorithm as its congestion control mechanism. Flows were injected in the bottleneck link incrementally every after 100 s simulation time. It can be observed that from 100 s to 200 s wherein a single flow of traffic is entering the link, less were the number of retransmission timeout occurrence for both protocols. In addition, both protocols frequently reached the upper window limit  $BDP_{ave}$  which indicates that the bandwidth allocation was maximized. It can be seen that the  $BDP_{ave}$



**Fig. 5.** RT-CaCC performance with varying cache sizes: (a) cache hits, (b) end-to-end delay, (c) throughput and (d) transmission cost

is always almost equal to  $B$  which ensures full buffer utilization. From 200 s to 300 s wherein an additional flow was injected in the bottleneck link, RT-CaCC registered fewer number of retransmission timeout than DTC. From 300 s to 400 s wherein three simultaneous traffic flows were entering the link, RT-CaCC



**Fig. 6.** Congestion window with increasing number of flows in a bottleneck link

obtained better bandwidth usage than DTC at high level of congestion. In addition, RT-CaCC congestion window is less aggressive than DTC which lead to better resource allocation and optimum cache utilization.

RT-CaCC was also evaluated with other cache-based transport protocols like DTC and ERCTP. These protocols also implement congestion control mechanisms but not cache-aware. Due to page limitation, the results of the comparison will be presented in the extended version of this work in another paper.

## 6 Conclusion

This work developed a cache-aware congestion control mechanism that uses cache management policies such as cache insertion and elimination policy as well as cache size allocation to mitigate packet losses from poor wireless link and congestion in the network. RT-CaCC outperformed the baseline protocols in terms of cache utilization, end-to-end delay and throughput from 10% to 50% average improvement gain. This only shows that using a cache-aware approach can effectively respond to packet losses either due to poor channel link or congestion in the network. Further, limiting the lower and upper boundries of the congestion window during high level of packet losses guaranteed optimum usage of cache memories while preventing the network from overshooting. In the future, RT-CaCC can be evaluated in a more challenging network scenario as well as incorporating a cache-aware cross-layer approach with the routing protocol in WSN such as RPL.

**Acknowledgment.** The authors would like to acknowledge the support of the University of the Philippines Diliman and the Department of Science and Technology (DOST) through the Engineering Research and Development for Technology (ERDT) Program.

## References

1. Akyildiz, I., Vuran, M.C.: Wireless Sensor Networks. Wiley, New York (2010)
2. Dunkels, A., Alonso, J., Voigt, T., Ritter, H.: Distributed TCP caching for wireless sensor networks. In: Proceedings of the 3rd Annual Mediterranean Ad-Hoc Networks Workshop (2004)
3. Braun, T., Voigt, T., Dunkels, A.: TCP support for sensor networks. In: Fourth Annual Conference on Wireless on Demand Network Systems and Services, WONS 2007, pp. 162–169, January 2007
4. Sharif, A., Potdar, V.M., Rathnayaka, A.J.D.: ERCTP: end-to-end reliable and congestion aware transport layer protocol for heterogeneous WSN. Scalable Comput.: Pract. Exp. **11**(4), 359–372 (2010)
5. Paek, J., Govindan, R.: RCRT: rate-controlled reliable transport protocol for wireless sensor networks. ACM Trans. Sens. Netw. **7**, 20:1–20:45 (2010)
6. Alipio, M.I., Tiglao, N.M.C.: Analysis of cache-based transport protocol at congestion in wireless sensor networks. In: 2017 International Conference on Information Networking (ICOIN), pp. 360–365, January 2017

7. Alipio, M.I., Tiglao, N.M.C.: Improving reliable data transport in wireless sensor networks through dynamic cache-aware rate control mechanism. In: 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), October 2017 (to appear)
8. Tiglao, N.M.C., Grilo, A.M.: Cross-layer caching based optimization for wireless multimedia sensor networks. In: 2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 697–704, October 2012
9. Tiglao, N.M.C., Grilo, A.M.: An analytical model for transport layer caching in wireless sensor networks. *Perform. Eval.* **69**(5), 227–245 (2012)
10. Meneses, D., Grilo, A., Pereira, P.R.: A transport protocol for real-time streaming in wireless multimedia sensor networks. In: 2011 7th EURO-NGI Conference on Next Generation Internet Networks, pp. 1–8, June 2011



# A New Lightweight Mutual Authentication Protocol to Secure Real Time Tracking of Radioactive Sources

Mouza Ahmed Bani Shemaili<sup>1</sup>, Chan Yeob Yeun<sup>2✉</sup>, Mohamed Jamal Zemerly<sup>2</sup>, Khalid Mubarak<sup>1</sup>, Hyun Ku Yeun<sup>1</sup>, Yoon Seok Chang<sup>3</sup>, Basim Zafar<sup>4</sup>, Mohammed Simsime<sup>6</sup>, Yasir Salih<sup>4</sup>, and Gaemyoung Lee<sup>5</sup>

<sup>1</sup> Computer Information and Science Division, HCT, Abu Dhabi, UAE  
`{malshemaili, kalhammadil, hyun.yeun}@hct.ac.ae`

<sup>2</sup> Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, UAE  
`{cyeun, jamal.zemerly}@kustar.ac.ae`

<sup>3</sup> School of Air Transportation and Logistics, Korea Aerospace University, Goyang, Korea  
`yoonchang@kau.ac.kr`

<sup>4</sup> Department of Electrical Engineering, Umm Al-Qura University, KSU, Mecca, Saudi Arabia  
`{bjzafar, ysali}@uqu.edu.sa`

<sup>5</sup> College of Engineering, Jeju National University, Jeju, Korea  
`myounglk@jejunu.ac.kr`

<sup>6</sup> Ministry of Hajj, KSU, Mecca, Saudi Arabia  
`msimsim@hajj.gov.sa`

**Abstract.** Radioactive applications are employed in many aspects of our life, such as industry, medicine and agriculture. One of the most important issues that need to be addressed is the security of the movement of radioactive sources. There are many threats that may occur during the transportation of the radioactive sources from one place to another. This paper investigates the security issues in the transportation of the radioactive sources. Thus, it is an attempt to build a secure, real time freight tracking system in which the radioactive source can be under inspection and control at all times during transportation from the shipment provider to the end user. Thus, we proposed a novel lightweight mutual authentication protocol to be used for securing the transportation of radioactive materials. Also, the security requirements for the proposed protocol were verified using the Scyther tool.

**Keywords:** Radioactive sources · Cyber security, mutual authentication  
Scyther tool · Real-time tracking

## 1 Introduction

Radioactive applications have grown rapidly in our lifetime as they can be used for a variety of purposes. For example, radioactive sources are used in the field of medicine for cancer therapy and blood irradiation. Additionally, these sources can be used in engineering to check flow gauges or to test soil moisture and material thickness/integrity

for construction or by specialists to irradiate food to prevent it from spoiling. However, radioactive sources can be dangerous in that they can negatively affect the user's health; thus, in order to overcome this problem, there is a need to instruct the user in how he/she can use these applications in a safe way [1, 2].

To handle these radioactive sources in a safe way, they must be controlled under a professional authority or organization. There are several organizations all over the world that regulate the usage of radioactive applications. The most famous of these organizations is the International Atomic Energy Agency (IAEA). The IAEA is the world's centre of cooperation in the nuclear field. It was set up as the world's "Atoms for Peace" organization in 1957 within the United Nations family. This agency works with its member states and multiple partners worldwide to promote safe, secure and peaceful nuclear technologies [3]. In the United Arab Emirates (UAE), there is a dedicated agency that addresses radioactive sources, known as the Federal Authority for Nuclear Regulation (FANR), which was established in 2009. FANR is the authority that is responsible for regulating all of the nuclear activities and licenses that involve the use of radioactive sources in UAE. FANR cooperates with the IAEA to satisfy international practice of the peaceful use of nuclear energy [4].

Using a radioactive source can lead to a disaster if the official oversight is insufficient. In this case, a poorly regulated source is known as a vulnerable source. A vulnerable source which safety cannot be ensured and which is not under regulatory control due to its being lost or stolen can become an orphan source. Both of these sources can carry risk levels that can lead to the damage of human health. The potential dangers can include injury or death. A serious example is the Teletherapy Heads accident, which happened on Samut Prakarn, Thailand in 2000, resulting in the deaths of three workers as a consequence of their exposure to the radiation [5].

In addition, orphan sources can be used in terrorist activities, such as using it to produce a radiological dispersal device (RDD). RDD can be used maliciously to produce a 'dirty bomb' or it can be spread deliberately and therefore expose innocent people to radioactive radiation. Terrorists routinely use this as a weapon to destroy the peace of the community by placing radioactive sources in public areas [6, 7].

To overcome the vulnerabilities that may occur during the transportation of the radioactive source, IAEA has come up with certain regulations and measures that should be considered during the design stage, such as measuring the quantity and the physical and chemical form of the radioactive material. Additionally, the mode of transport and the type of packaging used to transport the source must be taken into account.

We can add to these regulations the importance of identifying the threats and malicious acts that may occur during the transportation of the radioactive sources and provide rapid preventative counter measures to any unauthorized attempts to access the source. This is used to prevent, disrupt and defeat a terrorist operation before it occurs. Thus, one of the most important issues that require addressing is the security of the movement of the radioactive sources, as there is a significant threat that may occur during the transportation of these sources from one place to another.

This paper focuses on the security issues in the transportation of the radioactive sources. This security can be achieved by building a secure, lightweight (in terms of CPU workload) real time freight tracking system in which the radioactive source can be

under inspection and control at all times during its transportation from the shipment provider to the end user. The real time tracking system can be achieved by combining technologies from the Real Time Location Systems (RTLSs), which are the Global Positioning System (GPS) [8], Global System for Mobile Communications (GSM) [9], the 3rd Generation Project (3GPP) and active Radio Frequency IDentification (RFID) in the form of E-Seal [10], while the Wireless Sensor Network (WSN) [11] adds an environment sensing ability to the moving radioactive sources. However, combining all of these technologies can also give rise to certain security issues such as replay, modification, eavesdropping and Man-in-the-Middle attacks that the system needs to address.

The contribution of this paper is to address the security threats and secure communication of several entities in the real time tracking of radioactive transportation system and having them all synchronized and mutually authenticating each other. This is combined with formal methods verification proof and performance comparison with other protocols.

The main motivation of this paper is to design a secure real time freight tracking system. Thus, within the tracking system there is a need to secure the messages of transporting a shipment in a vehicle from the shipment provider to the end user. Consequently, in order to send messages in secure manner there is a need to design a secure communication among the two communication parties.

Thus, this paper designed a new mutual authentication protocol and is organized as follows. Section 2 explains the radioactive sources transportation scenario. Then, the related work is clarified in Sect. 3. Section 4 explains our proposed mutual authentication protocol, which is used to secure the system communication messages during the transportation of the radioactive materials from source to destination. Section 5 discusses the security analysis of our proposed mutual authentication protocol. Finally, Sect. 6 concludes the paper.

## 2 Scenario of Radioactive Sources

E-Seal is an active RFID tag that is used to lock the containers of physical goods. If anyone tries to tamper or remove the E-Seal tag, it will send an alarm to alert the shipment provider that there could be a physical attack on this content. The E-Seal can then provide evidence of the authenticity and integrity of the goods as well as physical protection. For more information about E-Seal, the reader can refer to our previous contributions about securing an E-Seal real time tracking system for the Internet of things [12, 13]. In our system, the E-Seal uses the cryptography symmetric approach in [14], which depends on a shared secret key between the shipment provider and another trusted party at the end user side. Thus, nobody can unlock the truck until it reaches its destination (including the driver of the truck himself). The process of transporting a radioactive source shipment in a truck from a shipment provider to an end user consists of three main stages as follows:

1. At the Shipment Provider (Main Center)
2. From the Shipment Provider to the Destination
3. At the End User

Using the cryptography symmetric approach increases the protection of the truck [12, 13]. If any attacker tries to intercept the truck on its way to the end destination, he/she will not be able to unlock the E-Seal because the driver does not have the key that can unlock the truck. Thus, E-Seal can provide security to the truck until it reaches the final destination.

The next section explains the security threats that can occur during the transportation of the radioactive materials from the shipment provider to the destination.

### 3 Related Works

The application that we need to secure is a real time freight tracking system. Usually for tracking system we only need to authenticate the truck responses. However, the tracking of our system can have messages from the server to change the truck path. Thus, there is a need to design a mutual authentication protocol that authenticates both of the server and the truck. The SSL/TLS authentication protocol is deemed heavy due to required PKI infrastructure and instead the lightweight proposed protocol described in [14] is used. To the best of our knowledge no such end to end real-time tracking of radioactive sources systems exist in the literature.

As explained before the mutual authentication protocol that are discussed in this paper are related to the suggested solutions of the EPC networks since the designs of the real time freight tracking system is based on the EPC networks. Thus, this paper investigated the related work on the EPC protocols.

One of the most famous proposed protocols is that of Chien [15]. He proposed an ultra-lightweight mutual authentication protocol for RFID tags. The protocol consists of two main parts, which are the initialization phase and authentication phase. At the initialization phase, the server randomly chooses an initialization key  $K_i$  and initialization access key  $P_i$ . Additionally, each tag stores three values, which are the Electronic Product Code (EPCx) of the tag, the initial authentication key  $K_{i0}$  and the initial access key  $P_{i0}$ . After each successful authentication, the  $K_{i0}$  and  $P_{i0}$  keys are updated and stored as  $K_i$  and  $P_i$ . The server saves 6 values, which are: EPCx, the old authentication key  $KOLD$ , the new authentication key  $KNEW$ , the old access key  $POLD$ , the new access key  $PNEW$  and DATA that is related to information on each tag. However, this protocol is under replay attack which is presented in Peris-Lopez et al. [16].

Lo and Yeh [17] improved the Chien's protocol. Their tag stored five values, which are the authentication key  $K_i$ , the database access key  $P_i$ , the Electronic Product Code (EPCx), the transaction number (TID) and the last successful transaction number (LST). Both the  $K_i$  and  $P_i$  are generated from the PRNG function at the initialization time of the tag. The server side saved 12 values, which are the new authentication key  $KNEW$ , the old authentication key  $KOLD$ , the database access key  $P_i$ , the new Electronic Product Number Code (EPCx-NEW), the old Electronic Product Number Code (EPCx-OLD), the new fast search key (PRNG (EPCx-NEW)), the old fast search key (PRNG (EPCx-OLD)), the transaction number (TID) and the last successful transaction number (LST). Additionally, the server implements two functions, which are the binary string length function  $LEN()$  and the binary string truncation function  $TRUNC()$  that are used by

the server side to validate and authenticate the tag. Nevertheless, the protocol still suffers from tracing attacks [18, 19].

Yeh et al. [20] tried to improve the Chien's family protocols. With this protocol, the tag saves 4 values, which are the authentication key  $K_i$ , the access key  $P_i$ , the database index  $C_i$  set to  $C_i = 0$  at the start of the communication and the Electronic Product Code (EPC $_x$ ). The reader saves the Reader Identification number (RID) only and implements the hash function  $H(\cdot)$ , where the server saves 9 values, which are the new authentication key  $K_{NEW}$ , the old authentication key  $K_{OLD}$ , the new access key  $P_{NEW}$ , the old access key  $P_{OLD}$ , the new database index  $C_{NEW}$ , the old database index  $C_{OLD}$ , RID, EPC and DATA of each tag.

The protocol in [21] has two vulnerabilities, which are data integrity and forward secrecy problems; this is why Yoon [22] proposed an improvement to this protocol. The protocol adds XORing a session random value at the first message on the tag in order to provide forward secrecy. Additionally, at the server side, a new message called MAC =  $H(\text{DATA} \oplus \text{NR})$  is added to verify the message sent from the server side and to ensure the integrity of the data. However, this protocol still adds too much load on the server side and is vulnerable to tracking attacks.

Mohammad Ali et al. [23] noted that the Yoon's protocol is under data forgery, server and tag impersonation attacks. For this reason, this paper proposed an improvement to the protocol by having the tag generate two random numbers  $NT$  and  $NT'$  and change the way of computing the M1 and E messages. Unfortunately, this protocol still adds computational load on the server side. Additionally, the protocol is under desynchronization attack because the  $C_i$  index is updated whether the session is successful or not, where:  $C_i$  is corresponding to the database index stored in the tag to find the corresponding record of the tag in the database; thus, the attacker could then launch a false message just to change the  $C_x$  value on the server side so that it will not match the value sent from the tag side. Thus, we proposed a new lightweight mutual authentication protocol that provides fewer loads on the server and the tag than other protocols in Sect. 6.

## 4 The Proposed Lightweight Mutual Authentication Protocol

A new lightweight protocol is one that is dependent on light computation, such as Pseudo-Random Number Generator (PRNG), and simple functions, such as (CRC) checksum, but not hash functions. Our proposed protocol depends on the PRNG only, which is why it is considered as a lightweight authentication protocol. Additionally, our protocol reduces the load on the server side. The SSL/TLS authentication protocol is deemed heavy due to required PKI infrastructure and instead the lightweight proposed protocol described in [14] is used.

Notations: The following notations are used in the proposed protocol:

- $\oplus$  denotes XOR.
- $\parallel$  denotes concatenation.

- **TS:** Active RFID Tag and sensor installed on the container. Each container has its own TS. Here, for the sake of simplicity we assume that we have only one entity as TS.
- **E-Seal:** Active RFID and sensor used to lock the truck shipment.
- **CPU:** Central Processing Unit in the truck.
- **DB:** Database
- **PRNG:** The SGCA cipher stream algorithm proposed in [22] has two functions which are SG and CA. SG is used to produce PRNG stream to encrypt the exchanged messages on the communication implemented in TS, E-Seal, CPU and Server. Whereas, CA is used to update the keys used in the communication.
- **$K_{T1}$ :** Stored shared secret between the TS and the CPU. It is used to authenticate the TS to the CPU and stored in both of the TS and CPU.
- **$K_{T2}$ :** Stored shared secret between the TS and the CPU. It is used to authenticate the CPU to the TS and stored in both of the TS and CPU.
- **$K_{E1}$ :** Stored shared secret between the E-Seal and the CPU. It is used to authenticate the E-Seal to the CPU and stored in both of the E-Seal and CPU.
- **$K_{E2}$ :** Stored shared secret between the E-Seal and the CPU. It is used to authenticate the CPU to the E-Seal and stored in both of the E-Seal and CPU.
- **CD:** Data related to the container status provided from TS.
- **Information Packet (IP):** E-Seal has an information packet with a length of 49 bytes, which contains an EPC for each tag, E-Seal Serial Number (SN), timestamp (TS) and Cyclic Redundancy Check (CRC) calculated from IP to detect accidental changes within the tag data. Thus, **IP = CRC (EPC, SN, TS)**.
- **Meta-id<sub>T1</sub>:** A pseudonym randomly chosen for each TS. This value is stored in both the TS and the CPU used to identify the communication from the TS to the CPU.
- **Meta-id<sub>T2</sub>:** A pseudonym randomly chosen for the CPU stored at the both of the TS and CPU. This value is used to identify the communication from the CPU to the TS.
- **Meta-id<sub>E1</sub>:** A pseudonym randomly chosen for each E-Seal. This value is stored in both the E-Seal and the CPU used to identify the communication from the E-Seal to the CPU.
- **Meta-id<sub>E2</sub>:** A pseudonym randomly chosen for the CPU stored in both of the E-Seal and CPU. This value is used to identify the communication from the CPU to the E-Seal.
- **GPSC:** GPS coordinates saved in the CPU. This information is retrieved from the GPS receiver attached to the CPU.
- **$T_T$ :** Time of the truck retrieved from the GPS receiver attached to the CPU.
- **$K_{S1}$ :** Stored shared secret between the CPU and the Server used to authenticate the CPU to the Server and stored in both the CPU and the Server ( $K_1$  in the Server).
- **$K_{S2}$ :** Stored shared secret between the CPU and the Server used to authenticate the Server to the CPU and stored in both the CPU and the Server ( $K_2$  in the Server).
- **Meta-id<sub>S</sub>:** A pseudonym randomly stored in the CPU. This value is stored in both the CPU and the Server used to identify the records of the CPU in the DB.
- **$K_1$ :** Stored shared secret between the CPU and the Server used to authenticate the CPU to the database. The database stores the  $K_1$  [Old, New], which refers to old and new values.

- **K<sub>2</sub>**: Stored shared secret between the CPU and the Server used to authenticate the Server to the CPU. The database stores the K<sub>2</sub> [Old, New], which refers to old and new values.
- **Meta-id**: A pseudonym randomly stored at the Server. This value is stored in both the CPU and the Server used to identify the records of the CPU at the DB. The DB stores the Meta-id [Old, New], which refers to old and new values.
- **CV**: Stands for counter value saved on the server side, which has a value of 0 by default and can be increased by one with each successful communication between the CPU and the Server.
- **PATH**: This set the new path of the truck in case that the path needs to be change. At the start there is plan path between A and B that are saved in the truck so in case of emergency the truck can change its track based on PATH data that is received from the server.

The SGCA mutual authentication protocol is shown in Fig. 1. The protocol is used to secure the communication within the proposed real time tracking system. The truck has two types of messages, which are;

1. Messages sent from the TS
2. Messages sent from the E-Seal.

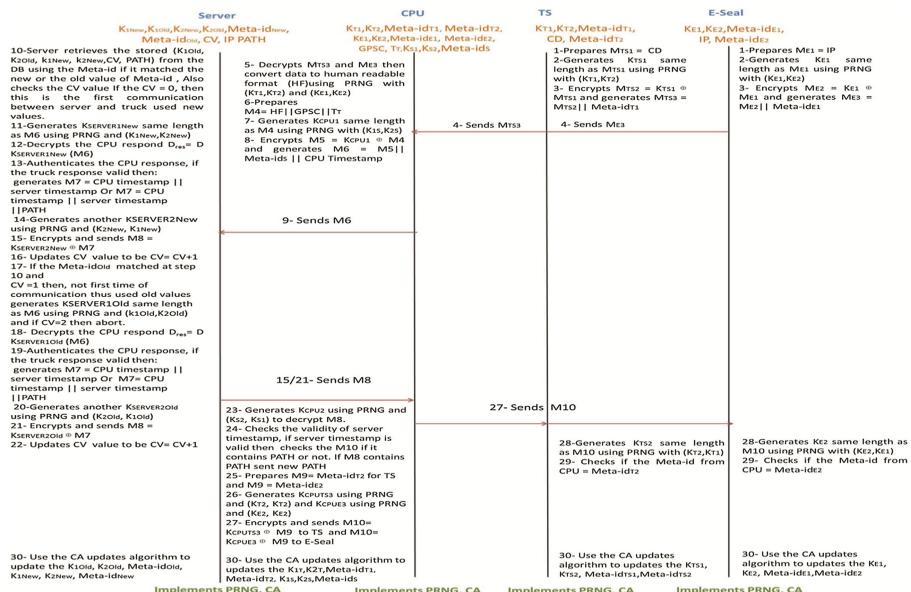


Fig. 1. The proposed mutual authentication protocol

As explained before, the truck has a reader that reads messages from the TS and E-Seal. The reader sends all these to the central processing unit (CPU) located in the truck. The proposed protocol assumes that the channel between the TS/E-Seal and the reader is insecure and the channel between the reader and the CPU is also not secure. The TS

sends encrypted messages regarding the status of the containers, while the E-Seal sends the Information Packet (IP).

CPU is a connector that connects the data sent from the container/E-Seal to the Central tracking Server and vice versa. Thus, in the proposed communication the CPU stored four shared keys between container TS/E-Seal and the CPU ( $K_{T1}$ ,  $K_{T2}$ ) for TS and ( $K_{E1}$ ,  $K_{E2}$ ) for the E-Seal. Also, another two shared keys between the CPU and the Server ( $K_{S1}$ ,  $K_{S2}$ ).

The CPU collects all of the messages received from the reader then decrypts using the shared secret key's ( $K_{T1}$ ,  $K_{T2}$ ) and converts them to a human readable language. Also, the CPU makes sure of the numbers of containers available in the truck and the status of each one. Also, the GPS receiver is attached to the CPU to acquire the GPS coordinates and time data. Thus, the CPU first prepares a message about the radioactive sources shipment status, quantities, GPS coordinates and time. Then, the CPU encrypts this message with the shared secret keys between the CPU and the Server ( $K_{S1}$ ,  $K_{S2}$ ) and sends it to the server using SMS through the GSM/GPRS/3G network. On the server side, there is a layer called the security center that is responsible for decrypting the SMS and then sending the data to the upper layer, as explained in Sect. 3. Thus, the communications in the SGCA protocol are between four main entities which are: TS, E-Seal, CPU and Server.

The protocol steps are as follows:

1. The TS and E-Seal first prepare their messages separately and send them at the same time. TS prepares  $M_{TS1} = CD$  where E-Seal prepares  $M_{E1} = IP$ .
2. The TS then generates a key using the proposed stream cipher (PRNG) called  $K_{TS1}$  that is used to encrypt the communication between the TS and the CPU. The PRNG uses ( $K_{T1}$ ,  $K_{T2}$ ) to generate  $K_{TS1}$ . The length of  $K_{TS1}$  is the same length of  $M_{TS1}$ . Also, The E-Seal generates a key using the proposed stream cipher (PRNG) called  $K_{E1}$ , that is used to encrypt the communication between the E-Seal and the CPU. The length of  $K_{E1}$  is the same length of  $M_{E1}$ .
3. The TS encrypts the message  $M_{TS2} = K_{TS1} \oplus M_{TS1}$ , generates  $M_{TS3} = M_{TS2} || Meta-id_{T1}$ . Also, The E-Seal encrypts the message  $M_{E2} = K_{E1} \oplus M_{E1}$ , generates  $M_{E3} = M_{E2} || Meta-id_{E1}$
4. The TS sends  $M_{TS3}$  and the E-Seal sends  $M_{E3}$  to the CPU through reader that will relay these messages to the CPU. For simplicity we did not add the reader as its job is only to relay the message from the TS/E-Seal to the CPU and vice versa.
5. The CPU gets  $M_{TS3}$  and  $M_{E3}$  from the reader and then retrieves the stored ( $K_{T1}$ ,  $K_{T2}$ ,  $K_{E1}$ ,  $K_{E2}$ ) from the DB using the  $Meta-id_{T1}$  and  $Meta-id_{E1}$  depending on the matched value. Then, CPU decrypts these messages using either PRNG with ( $K_{T1}$ ,  $K_{T2}$ ) for the TS or PRNG with ( $K_{E1}$ ,  $K_{E2}$ ) for the E-Seal. Also, the CPU converts these messages to a human readable format (HF). HF contains the information related to the status of the containers, E-Seal IP information and truck overall status, if the truck status is valid or not valid.
6. The CPU prepares  $M4 = HF || GPSC || T_T$ .
7. The CPU then generates a key using the proposed stream cipher (PRNG) called  $K_{CPU1}$  that is used to encrypt the communication between the CPU and the Server.

- The PRNG uses  $(K_{S1}, K_{S2})$  to generate  $K_{CPU1}$ . The length of  $K_{CPU1}$  is the same length of M4.
8. The CPU encrypts the message  $M5 = K_{CPU1} \oplus M4$ , and generates  $M6 = M5 || Meta-id_s$ .
  9. The CPU sends M6 to the Server.
  10. The server gets M6 from the CPU and then retrieves the stored  $(K_{1Old}, K_{2Old}, K_{1New}, K_{2New}, CV, PATH)$  from the DB using the Meta-id depending on the matched value if it is New or Old. Also, the server checks the CV value. If the CV = 0, then this is the first communication between the server and the truck and the New values of the Keys need to be used.
  11. The server generates key  $K_{S1New}$  using the PRNG algorithm and the shared  $(K_{1New}, K_{2New})$  between the CPU and the server with the same length of the received M6.
  12. The server decrypts the CPU response  $D_{res} = D K_{S1New} (M6)$ .
  13. The server authenticates the CPU response by checking the serial number SN of the packet on the DB and the validity of the timestamp; this information is then stored in the IP in the database. If the timestamp is valid and SN is correct, then the server generates  $M7 = CPU \text{ timestamp} || server \text{ timestamp}$ . If there is a need to set a new path to the track then  $M7 = CPU \text{ timestamp} || server \text{ timestamp} || PATH$ .
  14. Next, the server generates another  $K_{S2New}$  using PRNG, but this time it switches to  $(K_{2New}, K_{1New})$  as input to the PRNG algorithm), which will produce another stream cipher to encrypt M7.
  15. The server encrypts the message  $M8 = K_{S2New} \oplus M7$  and sends it to the CPU.
  16. The server updates the CV value to be  $CV = CV + 1$ .
  17. If at step 10 the matched value of the Meta- id is Old and  $CV = 1$ , then it means that this is not the first communication between this CPU and the server; thus, the server uses the old  $(K_{1Old}, K_{2Old})$  that are saved in the DB to generate  $K_{S1Old}$ . If the counter value = 2, then the communication is aborted as it may be a type of attack on the communication. Thus, we cannot use the old IVs more than twice.
  18. The server decrypts the CPU response  $D_{res} = D K_{S1Old} (M6)$ .
  19. The server checks again if the SN is correct or not and the validity of the timestamp. If it is correct, then the server generates  $M7 = CPU \text{ timestamp} || server \text{ timestamp}$ . If there is a need to set a new path to the truck then  $M7 = CPU \text{ timestamp} || server \text{ timestamp} || PATH$ .
  20. The server generates another  $K_{S2Old}$  using PRNG, but this time it switches to  $(K_{2Old}, K_{1Old})$  that will produce another stream cipher to encrypt M7.
  21. The server encrypts the message  $M8 = K_{S2Old} \oplus M7$  and sends it to the CPU.
  22. The server updates the CV value to be  $CV = CV + 1$ .
  23. The CPU receives M8 from the server and generates  $K_{CPU2}$  using PRNG and  $(K_{S2}, K_{S1})$  to decrypt M8.
  24. The CPU checks the validity of the server timestamp and, if the CPU timestamp is the same, it makes sure that the message is sent from the server. Also, in case of receiving a new path from the server the truck needs to set new truck path. If everything is satisfactory, then the truck moves to step 25; otherwise, the communication is closed.
  25. The CPU prepares  $M9 = Meta-id_{T2}$  for the TS and  $M9 = Meta-id_{E2}$  for the E-Seal.

26. The CPU generates another  $K_{CPU TS3}$  using PRNG, but this time it switches to  $(K_{T2}, K_{T1})$  that will produce another stream cipher to encrypt  $M9$  for the TS. Also, The CPU generates another  $K_{CPU E3}$  using PRNG, but this time it switches to  $(K_{E2}, K_{E1})$  that will produce another stream cipher to encrypt  $M9$  for the E-Seal.
27. The CPU encrypts and sends  $M10 = K_{CPU TS3} \oplus M9$  to the TS. Also, The CPU encrypts and sends  $M10 = K_{CPU E3} \oplus M9$  to the E-Seal.
28. The TS gets  $M10$  and generates  $K_{TS2}$  same length as  $M10$  using PRNG with  $(K_{T2}, K_{T1})$  to decrypt  $M10$ . Also, the E-Seal gets  $M10$  and generates  $K_{E2}$  same length as  $M10$  using PRNG with  $(K_{E2}, K_{E1})$  to decrypt  $M10$ .
29. The TS checks if the Meta-id from CPU = Meta-id<sub>T2</sub>. Also, E-Seal checks if the Meta-id from CPU = Meta-id<sub>E2</sub>. If it is correct then move to step 30 else close the connection.
30. All of the TS, E-Seal, CPU and Server use the CA update algorithm to update their stored such as TS, E-Seal use CA algorithm to update Keys and Meta-ids. Where, the server uses CA algorithm to updates the New Keys and Meta-ids, where the old value of the keys and Meta-id is the same as the new value of the final session.

## 5 Security Analysis of the Proposed Protocol

The proposed protocol can guarantee the mutual authentication requirements that were explained previously in the introduction, which are the secrecy, originality, freshness and integrity of the messages. The first requirement is the secrecy of the messages between the truck and the server, which can be achieved by encrypting messages between them using the proposed algorithm as explained before in Sect. 3. Additionally, the originality of the messages can be ensured because both the CPU and the server exchange secret keys between them, as well as encrypting/decrypting the messages that were relayed between them successfully. The same goes for the TS and E-Seal as the messages that are sending between them and the CPU are encrypted using another shared secret between the CPU and the TS, and the CPU and the E-Seal. Additionally, the protocol can ensure the freshness of the messages using the timestamp. The last requirement is the integrity, which can be achieved using the CRC (the CRC contained in the IP).

In addition, the protocol can resist the previously mentioned attacks, which can be divided into two parts as follows:

1. **Solutions to attacks on the truck used to transport the radioactive sources**, such as cloning, spoofing and physical attacks, are as follows:

**Cloning attack:** To prevent cloning attacks, the protocol uses in total six shared secret keys, which are two shared secret between TS and CPU ( $K_{T1}, K_{T2}$ ), two shared secret between E-Seal and CPU ( $K_{E1}, K_{E2}$ ) and two other shared secret keys between the CPU and the server ( $K_{S1}, K_{S2}$ ). Also, between the CPU and the server there is a timestamp, and between the CPU and TS/E-Seal there are agreed Meta-ids values. In M6, the CPU uses a timestamp that is encrypted using  $K_{CPU1}$  to authenticate the CPU. The server checks the validity of the timestamp and then replies with M8, which encrypts the timestamps of the CPU and the server so that the CPU makes sure that the message comes from the server and the server authenticates itself to the

CPU using the  $K_2$  as an encryption key. Also, the CPU authenticates itself to the TS/E-Seal using Meta-idT2 and Meta-idE2 that are stored in both of the CPU and TS/E-Seal.

**Spoofing attack:** The protocol can prevent this attack using the PRNG algorithm to encrypt the messages that are sent between the communication entities (TS, E-Seal, CPU and Server). Thus, even if the attacker captures the sent messages, he/she will not be able to recognize them.

2. **Solutions to attacks on the communication between the truck and the central tracking server.** These attacks can be the Tracing, desynchronization attack, Man in the Middle (MITM) attack and Replay attack as follows:

**Tracing:** The protocol resists tracing attacks using the Meta-id, which is not connected to the truck's ID. This can provide truck anonymity because the Meta-id is a pseudonym that is randomly chosen for each tag/WS/E-Seal. The value of the Meta-id is then updated after each successful authentication, which identifies the next communication between the CPU and the server.

**Desynchronization attack:** The attacker can launch a desynchronization attack on the CPU by increasing the CPU's Counter Value (CV) on the server side. However, the server specifies a time window for the validity of the CPU's CV such that the DB will accept the CPU only within the specific time window which is CV [0, 2]. Furthermore, the server stores both the previous and the new values that are used in the communication between the truck and the server. Consequently, this can be used as a backup plan that may help to increase the possibility of the system availability. Thus, even if the attacker is able to prevent the last message from updating new values (Steps 15/21), the protocol will still be running and DB will still be able to accept the CPU's old values.

**MITM attack:** First, the shared secret keys are exchanged in a secure manner. Thus, the four communicating parties can securely exchange the encrypted messages. In addition, the encryption of the messages can help to prevent the man in the middle attack. Additionally, the two parties use the challenge and response technique to ensure that the messages sent between the two parties are coming from legitimate parties and not modified by a third party because the message can be decrypted correctly by both parties.

**Replay attack:** The timestamp is also used to identify each protocol session between the CPU and the server. Thus, in this case, if the attacker is able to capture the previous message and wants to send it back to the server, and the timestamp is not sufficiently recent in the current time window, then the message will be rejected. Note that devices require secure synchronized clocks. Also, between the TS, E-Seal and the CPU there are agreed two Meta-ids that identify each communication between them. Also, the Meta-id values are updated after each successful authentication protocol.

In addition, the protocol can reduce the computational load on the server side using a Meta-id that identifies the truck in the backend database with O(1) computational complexity. The server can directly use the Meta-id to locate the corresponding entry

in its database and perform necessary computations for this matched entry only. Thus, the four communicating parties can securely exchange the encrypted messages. In addition, the encryption of the messages can help to prevent the man in the middle attack. Additionally, the two parties use the challenge and response technique to ensure that the messages sent between the two parties are coming from legitimate parties and not modified by a third party because the message can be decrypted correctly by both parties. The next subsection shows the protocol formal methods verification using the Scyther tool.

## 5.1 The Proposed Protocol Verification Proof with Scyther

To verify our protocol, we analyzed it using the protocol analyzer tool called Scyther [24]. Also, Scyther is a tool that can be used to analyze the security protocols under what is called the perfect cryptography assumption. The perfect cryptography assumption assumes that all cryptographic functions are perfect; for example, the attacker knows nothing from an encrypted message unless he knows the decryption key. The tool can be used to discover problems that arise from the way the protocol is assembled. In general, this problem is undecipherable, but in practice many protocols can be proven correct or attacks can be found. The main feature of the Scyther tool is that it is the only tool that can verify the synchronization [25]. The protocol synchronization means that the messages are transmitted exactly as was agreed upon by the protocol description.

The Scyther tool can verify the requirements of the mutual authentication protocol which are secrecy, integrity, fresh and originality. The first claim ensures the secrecy and integrity of the messages because the key used in the communication is secret, where the second claim ensures the originality of the messages. The third claim ensures the integrity as well because the communication between the two parties is continuous and they are able to answer each other with the encrypted messages. Additionally, the fourth claim shows that the two parties of the system communicate after receiving a message from each other. Thus, the availability and the freshness of the system can be guaranteed based on the fresh nonce that are used in the protocol model. Note that the default values provided by Scyther are used in the verification.

Figure 2 shows the results of verifying our protocol using Scyther. Our protocol passed all the claims.

Moreover, these results prove that the proposed protocol can be used to secure the communication in our system.

Table 1 shows a comparison between our protocol and the other protocols that are considered as lightweight mutual authentication protocols and described in Sect. 5. As shown in the table, our protocol provides all the security requirements, which are the ability to resist tracing, the replay attack, the desynchronization attack and the MITM attack. In addition, the protocol provides less computation load on the tag and server than other protocols using the Meta-id, which reduces the server load to  $O(1)$  instead of searching all the entries of the database.

MyProt	ESEAL	MyProt,eseal1	Secret nt1	<b>Ok</b>	Verified	No attacks.
		MyProt,eseal2	Secret nt2	<b>Ok</b>	Verified	No attacks.
		MyProt,eseal3	Secret IP	<b>Ok</b>	Verified	No attacks.
		MyProt,eseal4	Niagree	<b>Ok</b>	Verified	No attacks.
		MyProt,eseal5	Nisynch	<b>Ok</b>	Verified	No attacks.
		MyProt,eseal6	Alive	<b>Ok</b>	Verified	No attacks.
CPU		MyProt,cpu1	Secret nt1	<b>Ok</b>	Verified	No attacks.
		MyProt,cpu2	Secret nt2	<b>Ok</b>	Verified	No attacks.
		MyProt,cpu3	Secret nt3	<b>Ok</b>	Verified	No attacks.
		MyProt,cpu4	Secret HR	<b>Ok</b>	Verified	No attacks.
		MyProt,cpu5	Secret GPSC	<b>Ok</b>	Verified	No attacks.
		MyProt,cpu6	Niagree	<b>Ok</b>	Verified	No attacks.
		MyProt,cpu7	Nisynch	<b>Ok</b>	Verified	No attacks.
		MyProt,cpu8	Alive	<b>Ok</b>	Verified	No attacks.
SERVER		MyProt,server1	Secret HR	<b>Ok</b>	Verified	No attacks.
		MyProt,server2	Secret PATH	<b>Ok</b>	Verified	No attacks.
		MyProt,server3	Secret nt3	<b>Ok</b>	Verified	No attacks.
		MyProt,server4	Niagree	<b>Ok</b>	Verified	No attacks.

Done.

**Fig. 2.** Scyther verification proof for the proposed protocol**Table 1.** Comparison between the proposed protocol and others

	Chien [15]	Lo and Yeh [17]	Yeh [20]	Yoon [22]	Amin [23]	Proposed protocol
Resistance to tracking	No	No	Yes	Yes	Yes	<b>Yes</b>
Resistance to replay attack	Yes	Yes	Yes	Yes	Yes	<b>Yes</b>
Desynchronization attack	Yes	Yes	Yes	Yes	No	<b>Yes</b>
Resistance to MITM	Yes	Yes	Yes	Yes	No	<b>Yes</b>
Server load	3 CRC, 2 PRNG	8 PRNG, 8 CRC	7 PRNG, 1 Hash	8 PRNG, 1 Hash	13 PRNG, 1 Hash	<b>4 PRNG, 3 CA</b>
Reader load	1 PRNG	1 PRNG	1 PRNG, 1 Hash	1 PRNG, 2 Hash	1 PRNG, 2 Hash	-
Tag load	3 PRNG, 3 CRC	6 PRNG, 3 CRC	6 PRNG	6 PRNG	7 PRNG	<b>2 PRNG, 3 CA</b>

Furthermore, our protocol uses tag load less than the other protocols in the table with running 2 times PRNG, 3 times CA (Cellular Automata) [14] and 1 CRC for the E-Seal

only not for the TS (TS does not need CRC). Also, for the server side our protocol run the PRNG 4 times and CA for 3 times to update the keys and Meta-id new value which is considered less than the rest except Chien and Chen protocol that uses 2 times PRNG and 3 times CRC. Thus, in general we achieved our aims of reducing the computation load and providing security requirements. This also proves that our protocol is lightweight, as it does not consume as much computation as the other protocols that are compared in Table 1 based on Server load, Reader load and Tag load.

## 6 Conclusions

We have introduced a new lightweight mutual authentication protocol that can secure the communication involved in transporting radioactive materials. This paper analyses the threats that surround the transportation of the radioactive material and then emphasizes the importance of providing a mutual authentication protocol that can secure the communication during the transmission. Our protocol can resist the available attacks, such as the tracing, cloning, spoofing attacks, desynchronization attack, MITM attack and Replay attack. In addition, the proposed mutual authentication protocol is a lightweight solution that does not add too much computational load on the communication parties, such as server and tags because the protocol uses less PRNG calculations than the other lightweight protocols. Additionally, the proposed solution is strengthened by adding a robust PRNG to generate the keys of the communication.

The proposed mutual authentication protocol is verified using the Scyther tool, and the protocol has passed all five claims. Then, a simulation of the tracking system that implements the proposed protocol is provided to ensure its ability to secure the communication within the tracking system. The system used SMS messages to communicate with the size of the encrypted message, which is only 42 bytes. Additionally, the system can be customized to send information about the radioactive materials in a short time; thus, the data are sent very quickly between the communicating parties (for example, within 1.5 µs). In the future, we are going to implement our solution in a real life application. Also some future work will be done to evaluate the trade-off with other approaches such as: TPM (Trusted Platform Module), and SSL/TLS mutual authentication protocols. Also we would like to investigate if the designed protocol could be used in different scenarios.

**Acknowledgements.** The authors wish to acknowledge Information and Communication Technology Fund (ICT Fund) for the continued support for the educational development and research.

## References

1. Radioactive Sources: Uses, Safety, and Security, Australian Nuclear Science and Technology Organization, August 2016. <http://www.arpansa.gov.au/>
2. Sheldon, F.T., Walker, R.M., Abercrombie, R.K., Cline, R.L.: Tracking radioactive sources in commerce. In: WM 2005 Conference, Tucson, AZ, USA, 27 February–3 March 2005

3. Security of radioactive sources Interim guidance for comment, Printed by the IAEA in Austria, August 2016. [http://www-pub.iaea.org/MTCD/Publications/PDF/te\\_1355\\_web.pdf/](http://www-pub.iaea.org/MTCD/Publications/PDF/te_1355_web.pdf/)
4. Federal Authority for Nuclear Regulation, August 2016. <http://www.fanr.gov.ae/en/Pages/default.aspx>
5. Ya-Anant, N., Tiyapun, K., Saiyut, K.: Radiological accident and incident in Thailand: lesson to be learned. *Radiat. Prot. Dosim.* **146**(1–3), 111–114 (2011)
6. Security in the Transport of Radioactive Material, IAEA nuclear security series No. 9, Vienna, August 2016. [http://www-pub.iaea.org/MTCD/publications/PDF/Pub1348\\_web.pdf/](http://www-pub.iaea.org/MTCD/publications/PDF/Pub1348_web.pdf/)
7. Radiological Dispersal Device (RDD): Argonne National Laboratory, EVS, Human Health Fact Sheet, August 2005
8. Misra, P., Enge, P.: Global Positioning System: Signals, Measurements, and Performance, 2nd edn. Ganga-Jamun Press, Massachusetts (2010)
9. Schwieger, V.: Positioning within the GSM network. In: Proceedings on 6th FIG Regional Conference, San Jose, Costa Rica, 12–15 November 2007
10. Hunt, V., Puglia, A., Puglia, M.: RFID: A Guide to Radio Frequency Identification, 1st edn. Wiley, Hoboken (2007)
11. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. *IEEE Commun. Mag.* **40**(8), 102–114 (2012)
12. Shemaili, M.B., Yeun, C.Y., Mubarak, K., Zemerly, M.J., Chang, Y.S.: Securing E-seal real time tracking system for internet of things. In: Proceedings of 8th International Conference on Internet Technology and Secured Transactions, 9–12 December 2013, London, UK, pp. 65–69 (2013)
13. Yeun, C.Y., Shemaili, M.A.B., Zemerly, M.J., Mubarak, K., Yeun, H.K., Chang, Y.S.: ID-based secure real-time tracking system. *Int. J. Adv. Logist.* **4**(2), 100–114 (2015)
14. Yeun, C.Y., Shemaili, M.A.B., Mubarak, K., Zemerly, M.J.: A new lightweight hybrid cryptographic algorithm for the internet of things. In: Proceedings of 7th International Conference on Internet Technology and Secured Transactions, 10–12 December 2012, London, UK, pp. 87–92 (2012)
15. Chien, H.Y.: SASI: a new ultra-lightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Trans. Dependable Secur. Comput.* **4**(4), 337–340 (2007)
16. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J., Ribagorda, A.: Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard. *Comput. Stand. Interfaces* **31**(2), 372–380 (2009)
17. Lo, N.W., Yeh, K.-H.: An Efficient Mutual Authentication Scheme for EPCglobal Class-1 Generation-2 RFID System. In: Denko, M.K., Shih, C.-s., Li, K.-C., Tsao, S.-L., Zeng, Q.-A., Park, S.H., Ko, Y.-B., Hung, S.-H., Park, J.H. (eds.) EUC 2007. LNCS, vol. 4809, pp. 43–56. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-77090-9\\_5](https://doi.org/10.1007/978-3-540-77090-9_5)
18. Chen, C.L., Deng, Y.Y.: Conformation of EPC class 1 generation 2 standards RFID system with mutual authentication and privacy protection. *Eng. Appl. Artif. Intell.* **22**(8), 1284–1291 (2009)
19. Habibi, M.H., Gardeshi, M., Alaghband, M.R.: Practical attacks on a RFID authentication protocol conforming to EPC C-1 G-2 standard. *J. Ubicomp* **2**(1), 1–13 (2011)
20. Yeh, T.C., Wang, Y.J., Kuo, T.C., Wang, S.S.: Securing RFID systems conforming to EPC class 1 generation 2 standard. *Expert Syst. Appl.* **37**(12), 7678–7683 (2010)
21. Safkhani, M., Bagheri, N., Sanadhy, S.K., Naderi, M.: Cryptanalysis of improved Yeh et al.'s authentication protocol: an EPC class-1 generation-2 standard compliant protocol, IACR Cryptology ePrint Archive, Report. 2011/ 426 (2011)

22. Yoon, E.J.: Improvement of the securing RFID systems conforming to EPC class 1 generation 2 standard. *Expert Syst. Appl.* **39**(12), 1589–1594 (2012)
23. Mohammadali, A., Ahmadian, Z., Aref, M.R.: Analysis and improvement of the securing RFID systems conforming to EPC class 1 generation 2 standard, IACR Cryptology ePrint Archive, pp. 66–76 (2013)
24. Cremers, C.J.F.: The Scyther tool: verification, falsification, and analysis of security protocols. In: Gupta, A., Malik, S. (eds.) CAV 2008. LNCS, vol. 5123, pp. 414–418. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-70545-1\\_38](https://doi.org/10.1007/978-3-540-70545-1_38), <http://www.cs.ox.ac.uk/people/cas.cremers/scyther/>
25. Cremers, C., Mauw, S., de Vink, E.: Injective synchronisation: an extension of the authentication hierarchy. *Theor. Comput. Sci.* **367**(10), 139–161 (2006). Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. *J. Mol. Biol.* **147**, 195–197 (1981)



# Fog Computing as a Critical Link Between a Central Cloud and IoT in Support of Fast Discovery of New Hydrocarbon Reservoirs

Andrzej M. Goscinski<sup>1,2(✉)</sup>, Zahir Tari<sup>2</sup>, Izzatdin A. Aziz<sup>3</sup>,  
and Eidah J. Alzahrani<sup>2</sup>

<sup>1</sup> School of Information Technology, Deakin University, Geelong, Australia  
[ang@deakin.edu.au](mailto:ang@deakin.edu.au)

<sup>2</sup> School of Science, RMIT University, Melbourne, Australia

<sup>3</sup> Universiti Teknologi Petronas, Seri Iskandar, Malaysia

**Abstract.** The overall process of discovering hydrocarbon traps, starting with geological exploration through to Seismic Data Processing (SDP) is very expensive and time consuming. In the real-world, the oil and gas production relies on how soon seismic data is computationally processed. The ability for an oil and gas company to perform seismic computation at higher speed within shorter time provides competitive advantage in the race to discover new hydrocarbon reservoirs. We are convinced that the current state of research in areas such as cloud computing, fog computing, and edge computing will make a major change. The goal of this paper is to present the first step towards the development of such a three-level system and show its feasibility in the context of a model for hydrocarbon exploration and discovery operation.

**Keywords:** Seismic Data Processing (SDP)  
Hydrocarbon exploration · Fog computing · Edge computing  
Cloud computing

## 1 Introduction

Seismic data gathered from the Hydrocarbon Exploration and Discovery Operation is essential to identify possible hydrocarbon existence in a geologically surveyed area. However, the discovery operation takes a long time to be completed and computational processing of the acquired data is often delayed. Hydrocarbon exploration may end up needlessly covering an area without any hydrocarbon traces due to lack of immediate feedback from geophysical experts. This feedback can only be given when the acquired seismic data is computationally processed, analysed and interpreted timely. Therefore, we propose application of cloud technology and map it on a comprehensive model of facilitate Hydrocarbon Exploration and Discovery Operation using data collection, pre-processing,

encryption, decryption, transmission, and processing. The model exploits the logical design of Seismic Data Processing (SDP) that employs distributed systems and processing, and the ability for geophysical experts to provide on-line decisions on how to progress the hydrocarbon exploration operation, at a remote location, practically in the world of Internet of Things (IoT).

Many researchers are convinced that Fog Computing, is becoming the next big wave in computing due to the strong demand from IoT markets. As researchers of service computing, we are surrounded by numerous hypes and myths but also real opportunities of Fog and Edge Computing. It is time that we should have a clear understanding of the differences between the concepts of Fog and Edge Computing, and the role of Cloud datacentres in the new Fog Computing paradigm. In this paper, with a focus from service computing point of view, we regard Fog Computing as a critical link between a central cloud and IoT, and try to clarify these concepts and their relationships. The problem is how to apply the recently acquired knowledge and skills in clouds, fogs and IoT in the oil and gas discovery. We will show our mapping of these three-part computing technology on a Seismic Data Processing (SDP) model.

The main contributions of this paper are:

- A specification of the Seismic Data Processing in the terms of IoT
- Presentation of comparison and contrasts of fog computing and edge computing against IoT
- An original mapping of the clouds, fogs, edges on the IoT of the Seismic Data Processing model.

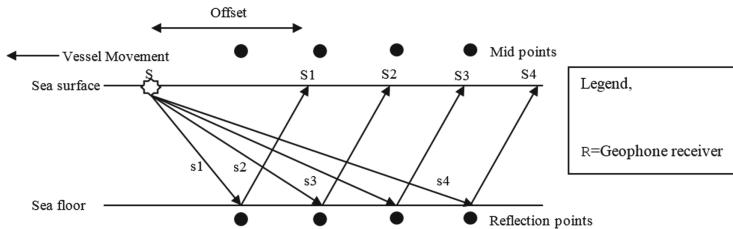
In this paper, Sect. 2 introduces basic concepts of hydrocarbon exploration and discovery their problems and solution requirements. Section 3 shows our model of Hydrocarbon Exploration and Discovery Operation, which demonstrates the IoT world of oil and gas exploration. Section 4 discusses and clarifies our approach to IoT and Clouds, in particular cloud, fog, and edge computing. Section 5 introduces our original mapping of clouds and fogs on IoT hydrocarbon exploration and discovery model. Section 6 concludes the paper.

## 2 Hydrocarbon Exploration and Discovery Operation Problems and Solution Requirements – IoT Perspective

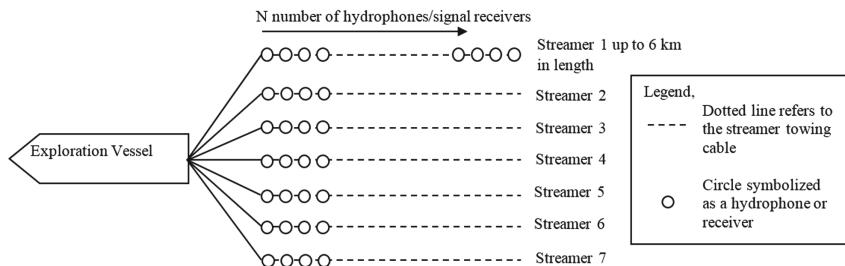
### 2.1 Data Collection

Seismic Data Processing depends on data collection. In a typical marine survey, exploration vessel tows airguns as sources of shocks or signals. The vessels also tow a stream of receivers or hydrophones to gather signals reflections. The seismic reflection data carries the properties of the Earth's subsurface as the propagated signals are reflected with different acoustic impedance levels (Fig. 1). Then, the seismic data is collected and stored at site on board exploration of vessels.

The layout of signal generators and receivers in a marine-based geological survey operation is shown in Fig. 2. So, during a marine based geological survey



**Fig. 1.** Data collection - signal reflection and middle points are at the same time state.



**Fig. 2.** Signal receiver deployment layout viewed from top.

operation, the  $50 \text{ km}^2$  of surveyed area is translated from a 200-km offset of a moving hydrocarbon surveying vessel by 0.25 km width. A single vessel can tow more than 7 streamers of receivers in a parallel layout. Each streamer reaches up to 6000 m in length and consists up to 480 receivers. The number of receivers depends on the distance interval set between individual receivers. For instance, if the receivers are set at an interval of 12.5 apart from each other over the 6000 m stretch, a total of 480 receivers can be towed in one streamer. If a vessel tows up to 7 streamers, this means that a total of 3360 receivers are available to record incoming signal reflections from multiple directions (Fig. 1). Simultaneous signal readings gathered from the receivers towed by the streamers construct a higher dimensional seismic data representation. Signal reflections captured by the receivers from multiple angles and directions enable the construction of a seismic data encompasses different orientation and dimensions. Since the area covered by each single vessel is large, and the number of vessels is big, the amount of data collected is huge.

## 2.2 Hydrocarbon Exploration and Discovery Operation Problems and Solution Requirements

Oil and gas discovery depends on interpretation of collected data and feedback passed on to the vessels. Feedback can only be given when the acquired seismic data is computationally processed, analysed, and interpreted. In this section, we

identify the issues of the current hydrocarbon exploration operation and a set of solution requirements, which form a basis of a model to address these issues.

### A. Problems Identification

From our expert interview in [1], we have identified four existing problems in relation to the current hydrocarbon exploration and discovery practice.

#### 1. Large Seismic Data Size

Seismic data acquired during a 3-month hydrocarbon exploration operation can yield up to 1 PBytes in size. Seismic data consists of signal reflection points resembling the Earth subsurface and formations. A small scale 122 GBytes of seismic data can contain as many as 24 million signal reflections points. The large size of seismic data contributes to problems such as transmission and processing times.

#### 2. Data Transfer

Seismic data are being transported in tape drives by helicopters and runner boats from exploration sites to private centralized processing centres on a fortnightly basis [1], due to two reasons:

- (a) High Value of Seismic Data - These data are very expensive and the oil and gas companies do not tolerate losing such valuable datasets through security breaches during transmission [1].
- (b) Data Communication - The current wireless network infrastructures used by the hydrocarbon industries from remote exploration sites to the processing centres is limited in terms of bandwidth and communication speed to transmit seismic data [1, 2].

Therefore, a conventional approach of manually transporting seismic data to centralized processing centres to carry out SDP is still preferred. However, the trustworthiness of human agent responsible for delivery of the seismic data to the processing centres is also questionable, because can be disclosed to the companys competitors during the delivery process. Nevertheless, the cost in terms of time loses for conventional data transportation from the remote exploration site to the designated processing centres is high.

The private processing centres possess state of the art HPC clusters [3]. Processing is carried out using commercial SDP software packages on these HPC clusters. Highly specialized commercial software packages for SDP are very expensive. According to [4], commercial SDP software packages are priced at \$3.15M (USD) for 5 licenses for a 5 year term.

#### 3. Computation Time

Seismic data acquired from the hydrocarbon exploration operation needs to be computationally processed to get a corrected signal reading. The computationally processed seismic data is interpreted by geophysical experts to identify any existence of a hydrocarbon reservoir. Computational processing consumes up to a few months when executed on a cluster computer or high-end machines [5]. According to [1], a computational time of one month is required to process 1 PBytes of industrial scale seismic data on a large HPC cluster, consisting of 128 nodes, with 1024 cores of processors.

#### 4. Processing Cost

The cost of processing the data is high. The current cost of SDP in private processing centres is approximately \$2k per km<sup>2</sup> seismic data [5]. A small-scale data acquired during a 1200 km<sup>2</sup> hydrocarbon exploration operation costs approximately \$2.4M. The SDP cost for 1200 km<sup>2</sup> can reach up to \$10M, depending on the complexity and granularity of the data.

### B. Solution Requirements

In response, we propose a set of solution requirements to facilitate the Hydrocarbon Exploration and Discovery Operation.

#### 1. Data Transmission

The remote location of hydrocarbon exploration sites in the middle of the sea make it impossible to be linked using wired network. A satellite network [8] is opted for a real time transfer to allow ubiquitous SDP from remote exploration sites to the center. Satellite services are reasonably inexpensive when trading off with the urgency to transmit and process seismic data. An average cost for a commercial business package intended for a dedicated satellite transmission speed of 1 Gbps is approximately a \$4,6k per month [6,7].

#### 2. Data Security

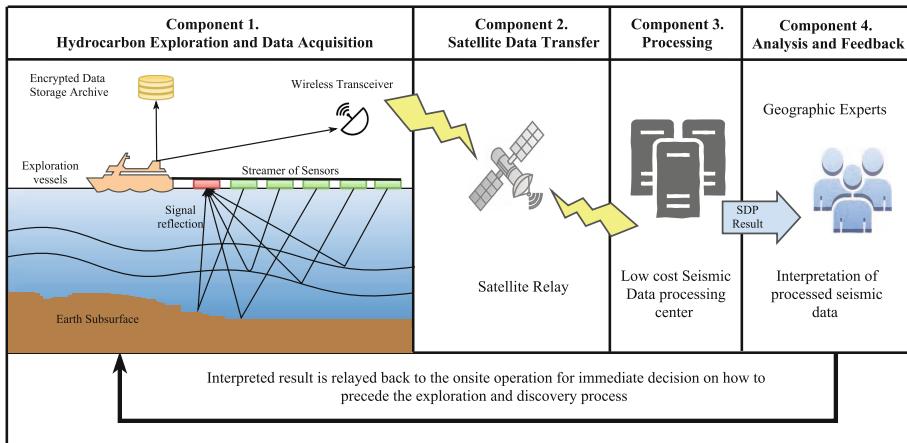
Seismic datasets are valuable due to the high operational cost and the potential of hydrocarbon existence presented in the datasets. Data transfer over the wireless network is highly subjected to data stealing and eavesdropping. A natural way of securing data before transmitting over the network is through data encryption. A fast encryption method is necessary to allow huge seismic datasets to be encrypted in a short time.

#### 3. Cost

Minimizing cost in hydrocarbon industries is a priority. Although hydrocarbon industries appear able to afford the expensive computing infrastructure and software packages, it is always imperative to find ways to minimize cost. In hydrocarbon exploration and discovery, costs can be reduced using cloud and much cheaper open source SDP software packages and higher processing capability providing outcome in a shorter time [8].

## 3 Hydrocarbon Exploration and Discovery Operation Model

Having defined the problems and solution requirements in the current hydrocarbon exploration and discovery operation, there is a need for a model to address these problems and solution requirements. In this section, we present our proposed model of hydrocarbon exploration and discovery from data acquisition and satellite data transmission through to data processing and feedback. The model addresses the problems listed in Subsect. 2.2.A and satisfies the solution requirements in Subsect. 2.2.B. Figure 3 shows the general idea of the proposed hydrocarbon exploration and discovery operation model [9].



**Fig. 3.** The model of Hydrocarbon Exploration and Discovery Operation using CBS.

The model consists of four components. They depict the operational sequence of seismic data acquisition, wireless transmission via the satellite to relay the acquired seismic data, the processing of seismic data in a low cost seismic data processing centre, and interpretation of the processed results by geophysical experts. The results are then relayed back to the onsite operation for immediate decision on how to proceed with the exploration and discovery operation.

### Component 1: Hydrocarbon Exploration and Data Acquisition

The first component of the model comprises two sub-components that take place in the remote hydrocarbon exploration sites.

#### 1. Data Collection and Storage

Seismic data is continuously gathered from the acquisition process and stored on disk that resides at the remote exploration site. Marine hydrocarbon exploration involves generating acoustic signal reflecting through the Earth subsurface, which are gathered by a stream of signal receivers at the surface. Signal reflection travelling times are recorded and represented as seismic traces on a data collection unit on the exploration vessel to be stored in a storage archive. Seismic data are gathered in a raw SEG-D format prior to transmission and later converted to a specific software package format for processing. SEG-D is the recommended seismic data format by the Society of Exploration Geophysicists (SEG) for newly acquired data from the hydrocarbon exploration operation [10].

Periodic transmission takes place when a threshold of approximately  $50\text{ km}^2$  block has been covered. The  $50\text{ km}^2$  approximation of geological survey is a representative value agreed by the hydrocarbon exploration contractors

to yield significant geophysical results when performing SDP [1,11,12]. The 50 km<sup>2</sup> of surveyed area is translated from a 200-km offset of a moving hydro-carbon surveying vessel by 0.25 km width. An area of 50 km<sup>2</sup> can be surveyed with a vessels speed of 6–10 knots or 11–18 kmh. The representative seismic data size encrypted prior to transmission.

## 2. Data Encryption

A natural way of securing data to allow transmission across the globe is through encryption. In our design, as soon as the whole 50 km<sup>2</sup> block of data is acquired, an encryption process is performed. A symmetrical encryption method with high bit key is considered to provide fast encryption with high security [13]. The 50 km<sup>2</sup> of a geologically surveyed area can yield up to 10 GBytes of seismic data. A high end system is commonly placed at the exploration site for data collection and pre-processing [7]. The process of encrypting this size of data requires a high-end server with at least quad core processors on board of the exploration vessel. A symmetrical encryption method [14] through a high-end system of 3 GHz quad core processors can encrypt a 10 GBytes of data at a computational speed of 60 s [15].

## **Component 2: Satellite Data Transfer**

The encrypted block of seismic dataset located on the remote data collection unit is now ready to be transmitted via a satellite network to the data processing centre for processing and analysis. The second component includes the transmission protocol and follows standards for the satellite data transfer.

### 1. Transmission Protocol

High bandwidth satellites offer natural support for communication mobility to the Internet across the globe. The Transmission Control Protocol (TCP) has been proven to support reliable Internet communication over the satellite [16]. A proven application that leverages on the TCP protocol by transmitting data using the Internet broadband over the satellite network is the digital TV broadcasting service. This service uses fast satellite transmission of a theoretical maximum bit rate or transmission speed of 1 Gbps to relay large stream of data [17]. A similar concept is useful to apply in transmitting a large amount of seismic data used in the hydrocarbon industry.

### 2. Data Transfer via Low Earth Orbiting (LEO) Satellite Network using Ka-Band Frequency

The breakthrough in satellite communication through the implementation of the Ka-Band frequency has made it possible to transmit large volume of data gigabytes in size. Ka-Band is a high resolution and focused microwave beam, which falls between the frequency ranges of 27.5 GHz and 31 GHz, initially used in military satellites, but has recently being commercialized. LEO satellite networks have been used to provide internet services on cargo and passenger vessels at a very high data transmission rate of 1.2 Gbps using the Ka-Band frequencies [18].

### 3. Wireless Standard: Worldwide Interoperability for Microwave Access (WiMAX)

Recent breakthrough research has demonstrated the applicability of using the WiMAX wireless communication standard operating between inter-satellites and mobile Earth transceiver stations [1]. The WiMAX IEEE standard 802.16 can transmit at a speed of 1 Gbps for up to 50 km in distance without signal amplifier or a repeater. Only a small bit error rate occurred when transmitting data beyond the distance of 50 km up to 400 km [20]. High bit rate data transfer with long range network propagation have championed WiMAX in the usage of satellite networking.

## **Component 3: SDP on Low Cost Data Processing Centre**

Seismic data gathered from the hydrocarbon exploration site is transmitted over satellite to the low-cost data processing centre. To minimize hydrocarbon exploration cost, the cost of processing seismic data needs to be significantly reduced.

The hydrocarbon industry does not need to acquire large processing facilities such as high-performance computers to perform seismic processing. The cost to maintain the computing infrastructure will be too high.

In our design, data processing centres are proposed to exploit clouds to perform SDP at a lower cost. Through clouds, computing infrastructure such as processors and storage can be leased out from the cloud providers, such as Amazon EC2 and Microsoft Azure. The concept of leasing computing infrastructure from the cloud providers released the burden from the hydrocarbon industry to pay for the overheads of maintaining the computing infrastructure.

Processing time for seismic data can be reduced significantly by adding more compute nodes and CPU cores [1]. Clouds computing technology offers scalable computing resources. On the other hand, a HPC cluster having only a fixed number of compute nodes is limited in terms of processing capability. Higher processing performance allow SDP to be executed in a shorter time [8].

An additional approach to reduce costs is using open source SDP packages, which incurs practically no cost. These SDP packages are installable on clouds [8]. Similar core seismic functions are available in both commercial and open source SDP packages. Commercial software packages contain seismic functions arranged in an integrated form featuring enhanced graphical layout. The enhancement of clouds to perform SDP will be discussed further in Sect. 5.

## **Component 4: Analysis Results**

The accelerated processing on clouds allows immediate analysis and feedback by geophysical experts even from across the globe. The processed data can then be analysed and interpreted for any possible hydrocarbon existence. Immediate decision and feedback can be delivered to the onsite remote hydrocarbon exploration location to proceed with the surveyed area, or otherwise refocused to another area, which can be more promising.

## 4 IoT and Clouds

### 4.1 Cloud, Fog, and Edge Computing

Many enterprises and large organizations begin to adopt the IoT, sets of small devices, sensors and actuators that provide services to users directly at the edge using networks, wireless networks, in general the Internet. Users deal with huge amount of data, big data, collected from edges and used to control the edges. To take advantage of them they must be transferred from/to distant, sometimes very distant sources of these data to be stored and processed by data and compute clouds. Since direct service links between clouds and edges, as it is in the case of discovery of new hydrocarbon reservoirs, do not allow accessing large amounts of data quickly enough, safety could not be guaranteed, reliability could be jeopardized, availability cannot be completely provided, and the whole system is subject to security attacks, there is a need for a further improvement.

Due to sizes of enterprises and large organizations, some smaller clouds, which could be not only stationary but also mobile, and servers are being proposed to improve these metrics. This is where the concept of Fog, Edge, and Mobile Fog and Edge Computing comes to play. Cloud Computing, Fog Computing, and Edge Computing, are crucial activities for many enterprises and large organizations, and a critical process for the IoT. Business Insider stated a couple of month ago, that nearly \$6T will be spent on IoT solutions in the following five years, and by 2020, 34 billion devices will be connected to the Internet, 24 billion within IoT [21]. Therefore, reaching this staggering figure depends just on both Fog Computing and Edge Computing.

There are examples of applications that due to data granularity collected and tasks allocated on the one hand, e.g., on a data collecting ship of a flotilla of oil and gas discovery ships in a city, county or a production division of a metallurgical plant, and finite power and storage capacity of (Mobile) Edge Clouds providing services on the other hand, require cooperation and coordination that has to be provided at the level that is lower than that provided by services at the Central Cloud level. These Central Clouds usually are located at fair distances from Edge Clouds. The problems of Edge Clouds and IoT devices shows up again, only at a higher abstraction level. And, this is where the concept of Fog, and Fog Computing comes to play the latest computing paradigm to support IoT applications, such as oil and gas discovery.

There are a lot of discussions and confusions around what is Cloud Computing, Grid Computing, Utility Computing, and Grid Computing 2.0. The history tells us it takes some time for people to really understand the differences and make clear definitions, but it did not prevent the advancing of the new technologies. In fact, these differences and definitions will become much clearer only after many successful applications and use cases have been produced. Therefore, in this paper, we are not trying to make solid definitions but rather than stimulate more discussions by proposing our own understanding of what are they and their relationships from the service computing point of view.

In the following, we will present a reference architecture for Fog Computing to further illustrate the concepts of Fog Computing and its relationships with Edge Computing and central clouds.

## 4.2 Fog and Edge Cloud Computing: Concepts and Relationships

The main confusion is the difference between Fog Computing and Edge Computing. We start with looking at the descriptions of Fog Computing and Edge Computing from major research venues.

IEEE Transactions on Service Computing Special Issues on Fog Computing [22]: The emerging Internet of Things (IoT) and rich cloud services have helped create the need for fog computing (also known as edge computing), in which data processing occurs in part at the network edge or anywhere along the cloud-to-endpoint continuum that can best meet user requirements, rather than completely in a relatively small number of massive clouds.

From the 1st IEEE International Conference on Fog and Edge Computing [23]: To satisfy the ever-increasing demand for Cloud Computing resources, academics and industry experts are now advocating for going from large-centralized Cloud Computing infrastructures to smaller ones, massively distributed at the edge of the network. Referred to as “fog/edge computing”, this paradigm is expected to improve the agility of cloud service deployments in addition to bringing computing resources closer to end-users.

From OpenFog Reference Architecture for Fog Computing by OpenFog Consortium [24]: Fog computing is a horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum. Fog computing also is often erroneously called edge computing, but there are key differences. Fog works with the cloud, whereas edge is defined by the exclusion of cloud. Fog is hierarchical, where edge tends to be limited to a small number of layers. In addition to computation, fog also addresses networking, storage, control and acceleration.

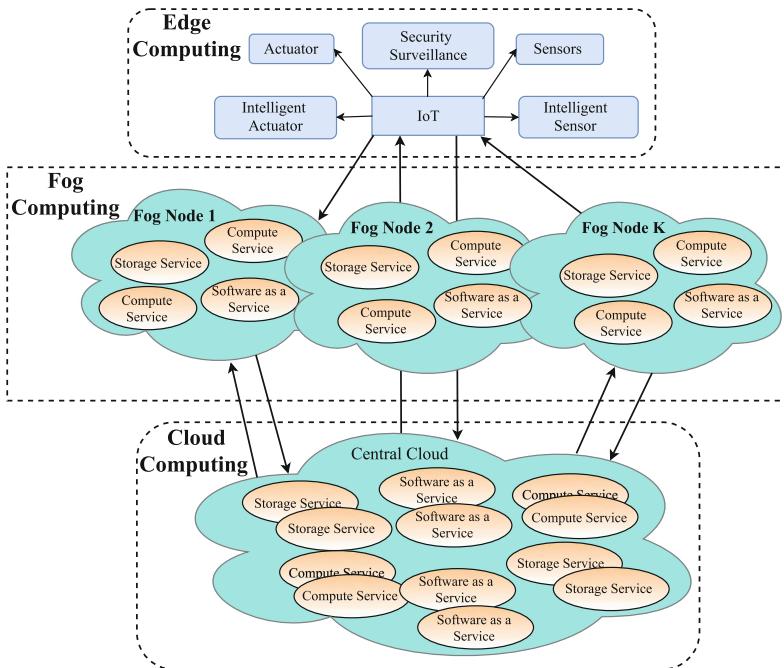
From the 1st IEEE International Conference on Edge Computing [25]: “Edge Computing” is a process of building a distributed system in which applications, computation and storage services, are provided and managed by (i) central clouds and smart devices, the edge of networks in small proximity to mobile devices, sensors, and end users; and (ii) others are provided and managed by the center cloud and a set of small in-between local clouds supporting IoT at the edge.

Apparently, there are two different views on the concepts of Fog and Edge Computing. Some researchers regard Fog and Edge Computing as the same paradigms with different names (as shown in [22,23]), while others distinguish between these concepts as two different things (as shown in [24,25]). We support the latter. Here we present our descriptions of Fog Computing as follows.

“Fogging” is a process of building a distributed system in which some application services, computation and storage, are provided and managed between Central Clouds and at the edge of a network in small proximity to mobile devices, sensors, and end users, by smart devices, even small local Edge Clouds, but others are still provided and managed by the in-between and/or Central Cloud; this

allows for Fog Computing. So, Fog Computing is a middle layer between the cloud and edge, hardware and software that provide specialized services.

The research on Fog Computing is to address the problem of how to carry out such a “fogging” process, e.g., how to manage the whole system, how to define and create fogs, provide Fog Computing (compute, store, communication) services. Many of these problems have not been defined yet, the whole Fog Computing is not defined; there are very many open problems.



**Fig. 4.** IoT World - Cloud, Fog, and Edge computing architecture.

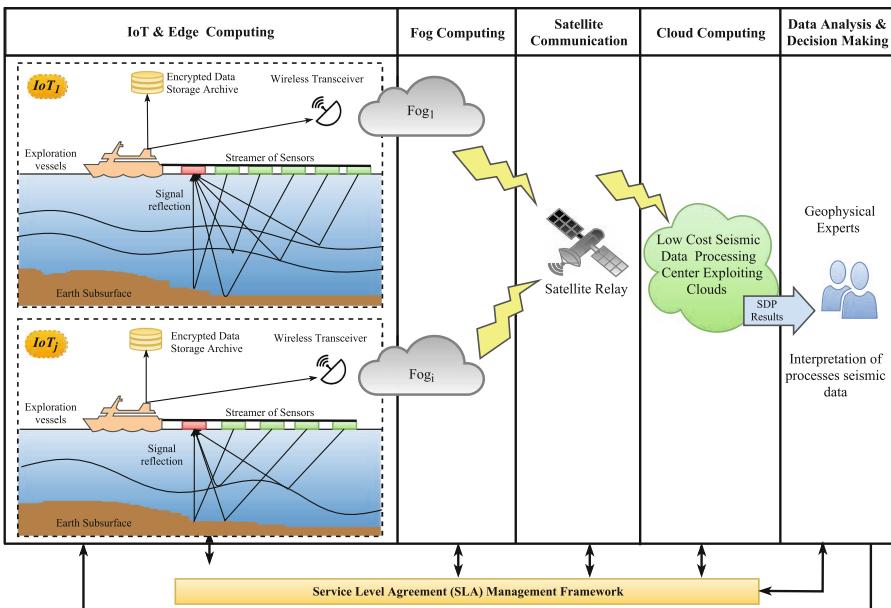
As shown in Fig. 4, Fog Computing, a part of the cloud stack, is the comprehensive computing paradigm that supports all sorts of IoT applications. Given the nature of different IoT applications and their requirements on computation resources such as compute, storage and software services, and their QoS constraints such as response time, security and availability, IoT applications may need to communicate with edge nodes only, or central clouds only, or both at the same time. Fog Computing can dynamically and seamlessly support all the three computing paradigms, viz. Edge Computing, Cloud Computing, and Fog Computing. Clearly, the major difference between Edge and Fog Computing is whether central clouds are included.

In summary, Edge Computing emphasis on processes at the “edge” and communication with the edge, while Fog Computing is carried out between the Central Cloud and the world of Edge Clouds, and thus includes the Edge. Fog

Computing services collaborate with and/or coordinate the cloud, the edge and the “world of IoT”, such as to play the roles of service providers, requesters, brokers, and so on. Thus, Fog Computing and Edge Computing have unique research topics as well as some overlapping topics. They form important subject area in research and practice. They are currently active and predictably booming soon. Their applicability in oil and gas discovery is discussed in Sect. 5.

## 5 Mapping Clouds and Fogs on IoT Hydrocarbon Exploration and Discovery Model

A description of the components of the Hydrocarbon Exploration and Discovery Model [8] shows methods used to identify potential locations of oil and gas and the four major components that were proposed to be designed, developed, and deployed using the technologies offered by Cloud Computing. However, a need for faster and cheaper discovery of oil and gas locations on the one hand and the development of new disruptive technologies in the areas of IoT, and Cloud and Mobile (Fog and Edge) Computing have generated an opportunity for the revision of mapping on these technologies on the Hydrocarbon Exploration and Discovery Model. The revised model is presented in Fig. 5. One of the most significant changes to the 2012 model is a new stage of processing based on the Fog Computing. We propose that a fog cloud could be deployed on one of the



**Fig. 5.** Mapping Clouds and Fogs on IoT exploration model.

vessels; fog computing carried out by this cloud is made responsible for dealing with big data and their fast encryption.

The innovative features of this model are: (i) application of IoT features; (ii) exploitation of Edge Computing supported by fast wireless transmission (G4 and G5); (iii) intelligent big data pre-processing; (iv) increasing processing power at the lower level of exploration data processing using Fog Computing supported by interconnected Fog Clouds; (v) application of stronger security countermeasures; and finally (vi) generation of faster feedback provided to individual vessels. All these features make oil and gas discovery faster, less expensive, more secure, and leading to making bigger profits.

The final achievement of this project is its validation of our cloud stack presented in Sect. 4, in particular making a clear distinction between Fog Computing and Edge Computing.

## 6 Conclusion

The IoT applications are acquiring data using different types of end devices (or Things) such as mobile phones, sensors, actuators, vehicles, and other devices. These end devices can talk to the Edge nodes that are extensions of the traditional network access nodes equipped with additional computing resources and server-side software services to handle the requests of end devices or push services and information to end devices. These edge nodes often need to work collaboratively to fulfil some service requested by moving objects such as people and vehicles. End devices can also talk directly to the central clouds that can provide much more diverse software services and unlimited computing resources.

Fog computing facilitates the computing continuum from end devices to the cloud. As for how far the continuum needs to reach, it is decided by the requirements of the applications. As shown in Fig. 4, the distance to the end devices is becoming farther and farther from the edge to the cloud, and they are connected through different communication network channels with different speed and bandwidth. Meanwhile, the processing power is becoming greater and stronger from the edge to the cloud. Therefore, normally if the applications require faster response time and less computation, fog clouds should be powerful enough to handle the service requests. However, if the applications require a lot of computation and the access to very large datasets, these service requests should be sent to cloud data centres either directly from the end devices or through the fog nodes after pre-processing. It should also be pointed out that in many cases the fog and central cloud can work between each other to optimize system performance and improve service quality.

In this paper, we have presented our views on the concepts of Fog Computing and its relationships with Edge Computing and Cloud Computing. The key point we want to emphasize here is to regard Fog Computing as a critical link between Central Clouds and IoT. We hope this paper could help to clarify some key concepts in Fog Computing, while in the meantime, stimulate more discussions and interests in the research and application of Fog Computing.

**Acknowledgments.** The authors wish to express their gratitude to Dr. Xiao Liu for making constructive comments to early versions of the paper. The authors would also like to thank the ARC (Australian Research Council) for the support of this project, under the Linkage scheme (LP#150101213).

## References

1. Ibrahim, N.A.: Expert Interview Session with Senior Geophysicist: Seismic Data Acquisition and Processing, Private Communication (2010–2012)
2. SpectrumData. The Seismic Data Graveyard (2011, 7 July 2011). <http://www.spectrumdata.com.au/content.aspx?cid=274>
3. Lefebvre, J.: Geocomputing Services and Solutions: Presentation of GSS services and Geocluster overview. CGGVeritas (2009)
4. Seismic Micro-Technology Inc.: Cost, Saving and Analysis Results. Seismic Micro Technology Kingdom Return on Investment (ROI) (2010, 7 July 2011). [http://www.seismicmicro.com/roi/roi\\_sample.htm](http://www.seismicmicro.com/roi/roi_sample.htm)
5. Technip: Oil and Gas Exploration and Production Reverse, Cost and Contract. Edition Technips, Centre Economic Paris (2007)
6. Ground Control: Satellite Internet Plans and Pricing: Satellite Internet Services From Ground Control (2011, 7 July). [http://www.groundcontrol.com/satellite\\_internet\\_service.htm](http://www.groundcontrol.com/satellite_internet_service.htm)
7. HughesNet: Broadband Satellite Plans and Pricing. Hughes Network Systems (2011)
8. Aziz, I.A., Goscinski, A.M., Hobbs, M.M.: Performance evaluation of open source seismic data processing packages. In: Xiang, Y., Cuzzocrea, A., Hobbs, M., Zhou, W. (eds.) ICA3PP 2011. LNCS, vol. 7016, pp. 433–442. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-24650-0\\_37](https://doi.org/10.1007/978-3-642-24650-0_37)
9. Aziz, I.A., Goscinski, A.M., Hobbs, M.: In support of hydrocarbon exploration and discovery using clouds. In: International Conference on Computer & Information Science (ICCIS), vol. 2. IEEE (2012)
10. Yilmaz, Ö.: Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data. Society of Exploration Geophysicists (2001)
11. Barakat, S.: Seismic acquisition system. U.S. Patent No. 7,573,782, 11 August 2009
12. Ashton, C.P., et al.: 3D seismic survey design. Oilfield Rev. **6**(2), 19–32 (1994)
13. Elminaam, D.S.A., Kader, H.M.A., Hadhoud, M.M.: Performance evaluation of symmetric encryption algorithms. IJCSNS Int. J. Comput. Sci. Netw. Secur. **8**(12), 280–286 (2008)
14. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999). [https://doi.org/10.1007/3-540-48405-1\\_34](https://doi.org/10.1007/3-540-48405-1_34)
15. Lafourcade, P., et al.: Symetric Encryption. Laboratoire Verimag Centre Equation, Private Communication (2008)
16. Arnal, F., Gayraud, T., Baudoin, C., Jacquemin, B.: Ip mobility and its impact on satellite networking. In: Advanced Satellite Mobile Systems, ASMS, pp. 94–99 (2008)
17. Shoji, Y., Takayama, Y., Toyoshima, M., Ohta, H.: Transparent transportation of digitized microwave environments over 10 Gbps optical networks: transportation of multi-channel digital broadcast signals. In International Conference on Transparent Optical Networks (ICTON), pp. 1–4, June 2010

18. Giffin, G., et al.: Satellite communications system for providing global, high quality movement of very large data files. U.S. Patent No. 7,783,734, 24 August 2010
19. Giuliano, R., Luglio, M., Mazzenga, F.: Interoperability between WiMAX and broadband mobile space networks [topics in radio communications]. IEEE Commun. Mag. **46**(3) (2008)
20. Zangar, N., et al.: Design, mobility management and analysis of a hybrid WiMAX/satellite network. In: IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications. IEEE (2009)
21. Here's how the Internet of Things will explode by 2020. [http://bit.ly/internet\\_of\\_things\\_forecasts\\_business\\_opportunities](http://bit.ly/internet_of_things_forecasts_business_opportunities)
22. IEEE Transactions on Services Computing - A Special Issue on Fog/Edge Computing and Services. [https://www.computer.org/cms/transactions/cfps/cfp\\_tscesi\\_fcs.pdf](https://www.computer.org/cms/transactions/cfps/cfp_tscesi_fcs.pdf)
23. 1st IEEE International Conference on Fog and Edge Computing, ICFEC 2017, Madrid, Spain, 14–15 May 2017. IEEE Computer Society (2017). ISBN 978-1-5090-3047-7. <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8014314>
24. OpenFog Reference Architecture for Fog Computing by OpenFog Consortium. <https://www.openfogconsortium.org/>
25. Goscinski, A.M., Luo, M. (eds.): Proceedings of 1st IEEE International Conference on Edge Computing, Hawaii, 25–30 June 2017



# Performance Assessment of Cloud Migrations from Network and Application Point of View

Lukas Iffländer<sup>1</sup>(✉) , Christopher Metter<sup>2</sup>, Florian Wamser<sup>2</sup>,  
Phuoc Tran-Gia<sup>2</sup>, and Samuel Kounev<sup>1</sup>

<sup>1</sup> Chair of Software Engineering, University of Würzburg, Würzburg, Germany  
[{ifflaender,kounev}@informatik.uni-wuerzburg.de](mailto:{ifflaender,kounev}@informatik.uni-wuerzburg.de)

<sup>2</sup> Chair of Communication Networks, University of Würzburg, Würzburg, Germany  
[{christopher.metter,wamser,trangia}@informatik.uni-wuerzburg.de](mailto:{christopher.metter,wamser,trangia}@informatik.uni-wuerzburg.de)

**Abstract.** Stateful migration processes for Cloud Services require the knowledge about their influencing parameters for the migration decision. Previous work focuses on the placement after the migration but not the migration process. In this work we evaluate the impact of network parameters on the migration performance as well as on the migrated applications. Therefore we propose an automatically set up testbed using OpenStack to measure key characteristics of the migration process.

## 1 Introduction

Cloud services are growing at a rapid pace and are expected to grow at an annual rate of 18% in 2017 in a 246.8 billion dollar market [1]. The reason for this trend is based on the promise of almost unlimited and scalable resources the cloud provides. Typically, cloud services are scaled and distributed in the cloud to meet demand and service requirements. This is directly supported by the cloud infrastructure by providing capabilities for orchestration, placement, and migration of cloud services in the virtual environments so that resources can be released or used upon request. *Service Migration*, the process of moving a service from one physical host to another, in particular, is an integral feature of the cloud to allocate resources at another host or location, and to adapt for the services accordingly. The cloud thus meets its economic expectations, as costs can be controlled and requirements of the services can be met if necessary.

There are many different migration types. For stateless services, migration is simply done by booting up a second instance of the service on the target host, switching the endpoint to the new instance and then shutting down the instance on the first host [2]. For stateful services migration is more complex and multiple approaches exist tailored to different requirements on factors such as migration speed and performance during migration. While these migrations provide many possibilities, many cloud and service providers fear the use of stateful migrations due to missing ways to predict their effect during the migration as well as the migration duration.

Many works present approaches on the topic where to place virtual machines respectively where to migrate these based on different performance characteristics [3–6] or on modelling the performance of virtualized systems [7–9]. Other work proposes models for VM migration times [10,11], propose a migration policy [12] or a migration progress management system [13]. Whilst these papers all introduce significant parts to plan and manage migrations, none provides detailed information on the used measurement methodology for the various migration phases and a detailed analysis of those measurements.

In this work we propose a testbed to analyze the impact of various parameters on the performance of migrations as well as on the performance effect of the migration on the migrated service. The testbed allows to measure the total migration time as well the migration phases. Using this testbed we benchmark migration processes under different network conditions (bandwidth, latency, drop rate) performing multiple migrations. Then we perform migrations of multiple applications under different condition simulating the migration inside a data center as well as the migration between data centers.

The contribution of this paper is

- the proposition of a testbed for migration benchmarking
- measurement and evaluation of the impact various network factors pose on migration duration
- measurement and evaluation of the impact the migration has on the migrated service under different network conditions

The remainder of the paper is structured as follows. In Sect. 2, related work is summarized and discussed. In Sect. 3, the background on cloud service migrations modes is outlined. The testbed is described in Sect. 4 whereas in Sects. 5 and 6, the evaluation is presented and results are discussed. Conclusions are given in Sect. 7.

## 2 Related Work

This section features work and research with the focus on the performance analysis of virtual machine migration within data centers.

In [14,15] overviews and reviews on the common techniques and open research questions of virtual machine live migration are given. While the former focuses on giving a comprehensive literature research and summing it up, the latter provides a comprehensive survey on VM migration schemes. After introducing aspects of migration, state-of-the-art live and non-live migration techniques are reviewed and investigated. The authors conclude by summarizing open research questions that require solving in order to optimize VM migration.

A model predicting the duration of virtual machine migration is presented in [10]. At first, the parameters influencing the migration performance, and their dependencies, are identified. With the aid of two simulation models, predictions with an accuracy within 90% are possible. This paper gives only limited information on how the results were measured and which environment was used.

Liu *et al.* propose a model to estimate the costs in terms of performance and energy consumption in [11]. Their approach is comparable to the previous work and analyzes the key parameters that impact virtual machine migrations. Their model is evaluated using workloads in a Xen virtualized environment, presenting results with higher than 90% prediction accuracy.

According to [12] current migration algorithms based on a single objective lack the consideration of factors influencing the migration process. Therefore, they propose a migration policy that considers the migration process as a multi-objective problem. Testing their policy using CloudSim, their results promise an increased system performance.

[13] addresses the issue that currently no state-of-the-art live migration progress management system exists. In the opinion of the authors multiple problems arise from this lack of management. For example, it is possible that the performance of application, which is distributed over multiple machines, is degraded, as a split with increased delay between these virtual machines, leading to a higher latency, could occur. Pacer, their approach to a migration progress management system, addresses these issues by relying on run-time measurements of various metrics, analytic models and on-the-fly adaptation. Their experiments on a local testbed and on Amazon EC2 promise a high efficiency. The problem of disimproved dependencies among virtual machines after migration is also raised by [6]. AppAware, their contribution on this topic, evaluates dependencies between guests, and the placement of them on the hardware hosts in order to optimize the migration process. Simulations show that their proposal can lead to a decrease of the network traffic by up to 81% in comparison to non-application-aware technique.

In contrast to the presented work, in this work we conduct measurements with the OpenStack platform and analyze the performance factors in regard to the overall migration duration and its parts.

### 3 Migration Modes

This section provides the essential background information on how the different cloud migration modes work. A common and crucial feature all cloud environments support is the migration of machines. This describes the process of moving virtual hosts and/or services from one physical host to another. This technique is required for multiple reasons. At first, if a host running multiple virtual machines requires maintenance, e.g. due to broken hardware or a scheduled software update, the host needs to be shut down or rebooted. Other use cases are the dynamic resource management for load or power balancing within a data center, and to seamlessly migrate virtual machines from a test environment into production. The simple solution, to just deactivate the host, and therefore also its running guests, is not feasible, especially on a commercial platform where customers pay for the availability of their product and service level objectives have to be met. Therefore, techniques enabling the movement of guests from one host to another, minimizing the downtime for the customer, are required. In the following, the most common migration types are introduced.

*Non-live migration.* This migration type is also known as cold or offline migration. At first, the availability of the required resources at the target host is validated. If enough resources are available, they are reserved on the target system. Now, the guest is shut down, then the virtual network of the guest is detached from the host, and the disk of the guest is moved to the target host. After completing this transfer, the virtual network connection is reattached at the target host and the guest system is restarted. This form of migration is noticed by the guest as a reboot.

*Live Migration.* The next approach is the so called Live Migration. Its goal is to migrate a machine without disconnecting client and application. Memory, storage and network connectivity are transferred between the hosts. Instead of rebooting, the machine is paused on the source host, the image is moved to the target host and there resumed. There are several approaches to live migration and to increase the migration speed and thereby reduce the service's downtime.

*Live Migration via Central Storage.* After the reservation of the resources on the target host a snapshot of the virtual machines memory is created and transferred while the machine is still operating on the source host. A shared central storage is mandatory for this type live migration. The guest machine is paused on the source host, its network connections are detached and the machine memory and its register content is transferred. Afterwards, the machine is resumed on the target host and its network connections are reattached. Depending on the transfer time of the remaining delta, the guest might only notice a sudden jump of the system time before and after the pausing of the machine.

*Block Live Migration.* The second approach is called block live migration. Block migration is a similar process to the before mentioned live migration. This time, no shared storage is required as the disk(s) of the guest are located on the compute hosts and therefore are also migrated. Thus, the total the data volume rises in comparison to the live migration.

*Pre-Copy Migration.* There are two ways to handle live migration. The first one is pre-copy migration. Since the machine is continuing its operation, the guest system and its memory most likely change during this transmission. Accordingly, the memory of the guest is compared to the already transferred content on the target host. If this delta is beyond a configured threshold, a new snapshot is created and transferred to the destination node. This process is repeated until the delta falls under a preset threshold. Then the machine is paused and the remaining delta is transferred. After resumption and reestablishment of the network connection the machine is fully operational without performance impact. The pre-copy migration can fail to ever complete if the threshold is set too low and/or the machine content changes too rapidly.

*Post-Copy Migration.* The secord way is the post-copy migration. Here the process starts with suspending the machine at the source host and transfer a

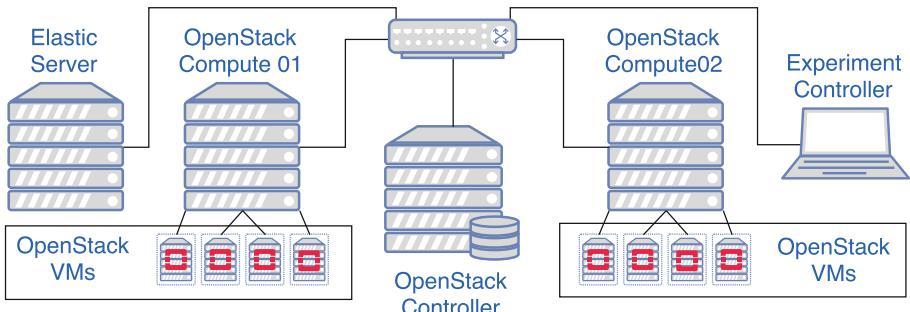
minimum of information (at least the register content) to the target host. The machine is then resumed at the target host and network is reattached. Then the remainder of the machine is transferred. During this phase page faults for not yet transferred pages are resolved over the network. The process is completed once the last fragments of the machine are transmitted. This process tackles the problem, pre-copy migration has with too rapidly changing content. On the other hand resolving page faults over the network can cause major performance problems.

*Nomenclature:* For nomenclature we orient ourselves at the OpenStack nomenclature, where *live migration* usually means *pre-copy live migration via central storage* and *block migration* is equivalent to *pre-copy block live migration*. These are also the two types we focus on in this paper.

## 4 Testbed Setup

Figure 1 presents the testbed used for the measurements presented in Sect. 5. A minimal OpenStack setup<sup>1</sup>, sufficient to create virtual machines with network access as well as perform block and live migrations has been installed. It consists of the two compute nodes 01 and 02 (SunFire X4150, Intel Xeon 5300, 16 GB RAM, 500 GB HDD) running the virtual machines, and the controller node (Fujitsu Esprimo C5730 E-Star 5.0, Intel Core2 Duo E8400, 4 GB RAM, 250 GB HDD) also running the storage service for the live migration. The experiment controller is used to configure, start, and stop the measurement runs. For result recording an installation of Elastic Stack is used. Experiment results generated from OpenStack log files are collected via LogStash and forwarded to an ElasticSearch server. All of these hardware devices are interconnected via an 1000 Mbit/s Ethernet switch.

Templates for virtual machines are called flavors in OpenStack. These flavors allow the user to define the amount of virtual CPUs, the memory, the



**Fig. 1.** Testbed setup

<sup>1</sup> <https://www.openstack.org/>.

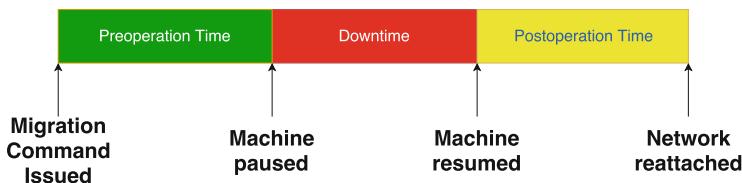
**Table 1.** The used OpenStack flavors and their properties

Flavor	VCPUs	RAM	HDD
m1.tiny	1	512 MB	1 GB
m1.small	1	2048 MB	20 GB
m1.medium	2	4096 MB	40 GB
m1.large	4	8192 MB	80 GB

disk size, and the network connection of the guest, just as when purchasing a hardware server. For the upcoming measurements we used four out of the five default OpenStack flavors: m1.tiny, m1.small, m1.medium m1.large. Their according properties are denoted in Table 1. Larger flavors where not explored due to memory requirements exceeding our testbed system.

## 5 Impact of Network Characteristics on the Migration Performance

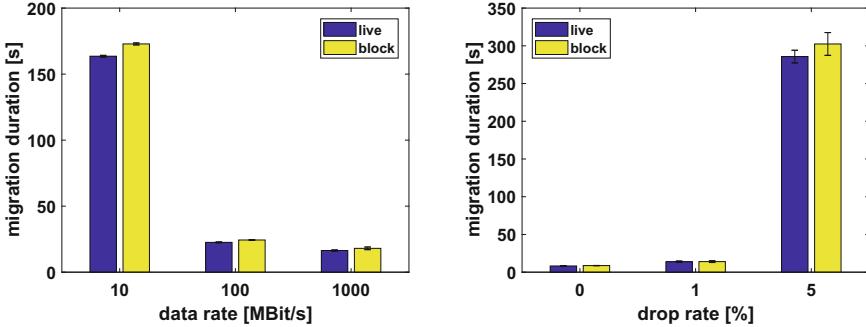
Figure 2 depicts the different stages of a migration that have been measured and evaluated. The migration time is composed of three blocks: *Preoperation Time*, *Downtime*, and *Postoperation Time*. The beginning of each migration is the execution of the migration command of a still running virtual machine. Now certain preoperations are executed until the machine is paused. This time is the *Preoperation Time*. Afterwards, the machines network is detached. The time until the machine is resumed on the target host is called the *Downtime*. Now the *Postoperation time* is running until the machines network is reattached. The relevance and portion of each step depends on the requirements of the migrated machine, or, accordingly, its service.

**Fig. 2.** Measurement parameters

In the following the results of the measurements taken for network characteristic influence are presented and discussed. Network settings are modified and the above specified parameters are measured in the scenarios to be presented.

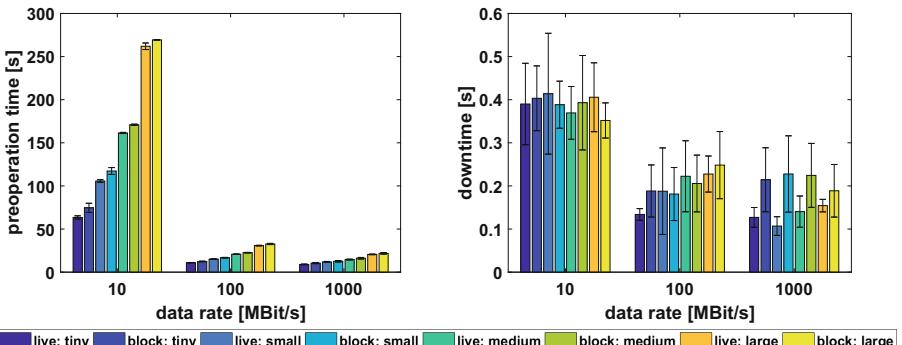
### 5.1 Impact of Network Throughput Limitations

A huge part of the migration time is allocated to the transfer of large amounts of data across the network. Therefore, migrations have been measured with bandwidth restriction of 1000 Mbit/s, 100 Mbit/s and 10 Mbit/s for the m1.tiny, m1.small, m1.medium and m1.large flavors.



**Fig. 3.** Migrations of the medium flavor using different throughput limits (left) and network drop rates (right)

Figure 3 shows the migration times for the medium flavor. The migration time increases when the throughput is reduced. These results are representative for the other flavors. Between 100 Mbit/s and 1000 Mbit/s the difference is quite small compared to the difference between 100 Mbit/s and 10 Mbit/s. The average migration times of the live migration approach are lower than for the block migration approach.



**Fig. 4.** Preoperation (left) and downtimes (right) for various flavors and migration types grouped by throughput

The majority of the total duration is spent on the *preoperation time* (Fig. 4). Therefore, it is also considerably affected by the throughput limit which relates to the copying of the first snapshot being located in the preoperation phase. The preoperation time is largely increased when reducing the available bandwidth. Live migration is faster than block migration. The downtimes are slightly sensible to the change of the throughput limits but the effect is weaker than for preoperation times since only a small part of the downtime phase relies on network transport. Live migration downtimes are always below the block migration downtimes by up to 50%. For the effect of the throughput limits on the *postoperation times* no statistical significant effect of neither the throughput limitation nor the migration mode can be found. The figure is therefore omitted.

## 5.2 Packet Loss in the Network and Related Effects

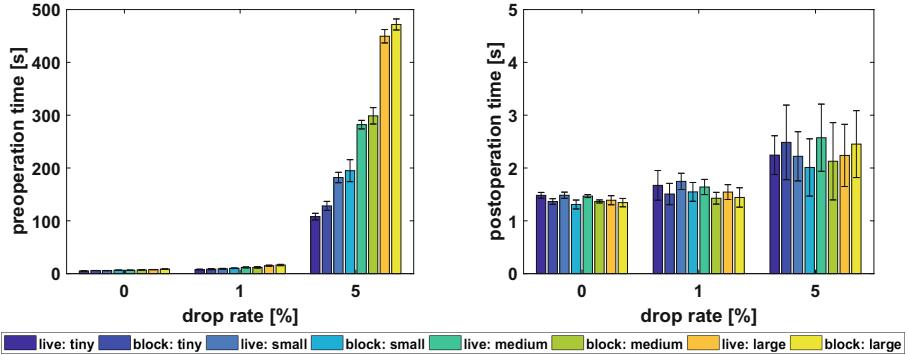
Another network parameter is the percentage of packets that are not successfully transferred e.g. due to uncorrectable bit errors in the line. This percentage is called the packet drop rate. The effect of the drop rate ( $d$ ) depends on the algorithms used by OpenStack's migration mechanism and the transport protocol [16].

OpenStack uses different approaches in different modules making use of both UDP or TCP for the migration as well as normal operation. Measuring the effect of different loss rates is required to estimate the impact of unstable links on the different subtasks of the migration duration. Thus, migrations have been measured without error as well as for one and five percent drop rate. The five percent setting has been chosen as the upper bound for realistic scenarios, representing an really unstable data link (e.g. for a compute center this could be a cellular fallback). In an ideal scenario with ideal protocols, the respective minimal increase factor in transfer duration would be 1.0101 (one percent) and 1.0526 (five percent) according to Formula 1.

$$n = \sum_{i=0}^{\infty} d^i \quad (1)$$

Figure 3 exemplarily shows the migration times for the medium flavor with different drop rates. The increase between no and one percent drop rate varies between 38% (m1.tiny) and 77% (m1.large) with block migration. The increase between one and five percent drop rate is even larger. The duration rises between 1164% (m1.tiny) and 2525% (m1.large). Any of these factors significantly exceeds the theoretical optimal factors that have been calculated above. This leads to the conclusion that a lot of packets are redundantly transferred and the link does not operate near optimal speed.

Independently of the drop rate, the live migration is slightly faster than block migration. While the relative advantage is constant, the absolute advantage increases with instance flavor and drop rate culminating at almost 23 s for the large flavor at five percent drop rate.



**Fig. 5.** Preoperation (left) and postoperation times (right) for various flavors and migration types grouped by drop rate

Again the *preoperation time* is the largest part of the total migration duration as depicted in Fig. 5. Similar to the duration, the live migration has a slight advantage over the block migration. The factor is even higher than for the total duration, leading to the assumption that the part influenced most by the packet loss is the transmission of the snapshot for the block migration and the synchronization of the network file system for the live migration. Ignoring singular exceptions, the average *downtime* increases slowly with the drop rate. For the first step only few scenarios (e.g. block with m1.medium) reach and exceed the factor of two. A slightly larger increase is visible at the second step with all flavors and modes at least doubling. On average live migration has a slightly better downtime but the confidence interval overlap. The behavior of the *postoperation time* is less sensitive to the increased drop rate. Figure 5 also shows the postoperation times. There is a slight increase between no and one percent drop rate and a slightly larger increase when increasing the rate to five percent. The increase is relatively small compared to the preoperation time at about sixty percent between zero and five percent drop rate. As for downtime, there is no significant difference between live and block migration.

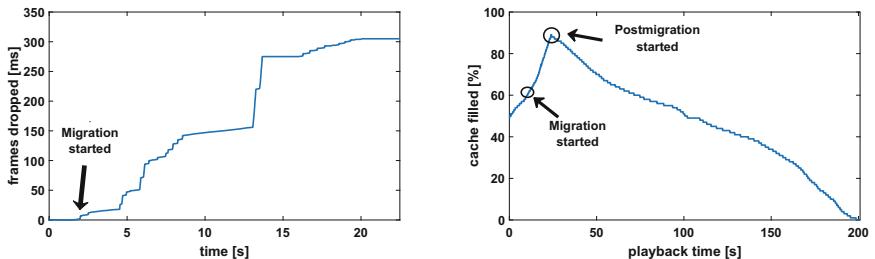
## 6 Evaluation of Migration Performance from Application Perspective

Applications have different requirements regarding the migration process. There are less time critical applications like downloads of large files. However, there are also more time critical applications with different requirements. Normal video streaming requires an acceptable quality with no stalling while live streaming requires a short transmission time to the user as well. The usability of some applications during migration has been tested to evaluate the effect of the migration on the application and ultimately the user's quality of experience (QoE) when using the application.

Three different approaches to video streaming were tested in two scenarios. *Scenario I* features no extra latency and a throughput of 1000 Mbit/s, which is similar to the environment parameters inside a data center. The migration at 100 Mbit/s and 100 ms delay in *Scenario II* resembles the migration over a mediocre link between remote data centers. For the video content the Sintel<sup>2</sup> movie has been chosen due to its licensing and its popularity. An application scenario is considered migrateable, when the migration successfully completes and no noticeable impact for the user is visible.

### 6.1 Server-Based Streaming Using Colocated Content

Live video streaming requires the parallel encoding and distribution of the material. Performing both tasks on a single machine puts high requirements on the CPU to achieve the encoding in real time. For this task, the widely utilized ffmpeg<sup>3</sup> was used. For the actual encoding, the ffmpeg application was used to convert the source video into a buffer file. For the streaming itself, ffserv was used to transfer the aforementioned buffer file. In this scenario, the streaming intelligence is supposed to be on the server side. Therefore, MPplayer<sup>4</sup> has been chosen for this task. MPplayer does neither automatically reconnect nor adapt the streaming quality. It just plays a video from a provided URL.



**Fig. 6. Left:** Frames dropped during the beginning of the migration process using server side streaming with co-located content in *Scenario I* **Right:** Percentage of buffer fill level over time before, during and after migration using server side streaming with external content in *Scenario I*

A server of the m1.large flavor is setup to evaluate the migrateability of this scenario. The large flavor is necessary to provide enough computing power to run the encoding since it provides four vCPUs.

After the migration is triggered in *Scenario I* some transmission errors occur when no cache is enabled. Leading to a large number of dropped frames as seen in Fig. 6. To the user this is visible as a stuttering playback of the stream

<sup>2</sup> <https://durian.blender.org/download/>.

<sup>3</sup> <https://www.ffmpeg.org/>.

<sup>4</sup> <http://www.mplayerhq.hu/>.

and sudden jumps of a few seconds ahead. If sufficient caching is enabled, this problem is not visible to the user. Only the cache level drops slightly. After most of the machine is migrated, the server migration is taking a lot of time for the remaining part. The log file shows that the content remaining to be transferred permanently increases. This is due to the fact that encoder permanently encodes the video and writes to the buffer file. This leads to a never-ending migration process. It is permanently stuck, alternating between zero and five percent of remaining content. Even after 20 min the migration process has not finished. As soon as the encoding is terminated remotely, it is only a matter of a few seconds and the migration finishes. The experiment has been repeated with *Scenario II*. This time, it takes longer to reach this loop state. Additionally, the video playback also stalls. The fact that the migration not only noticeably impairs playback quality but also never completes renders this application scenario *not migrateable*.

## 6.2 Server-Based Streaming of External Content

As the concept detailed in Subsect. 6.1 fails, the idea is to move the encoding to a separate node and migrate only the node running the streaming server. The used applications remain the same. With the required amount of processing power significantly reduced, it is possible to switch to the smaller m1.small flavor.

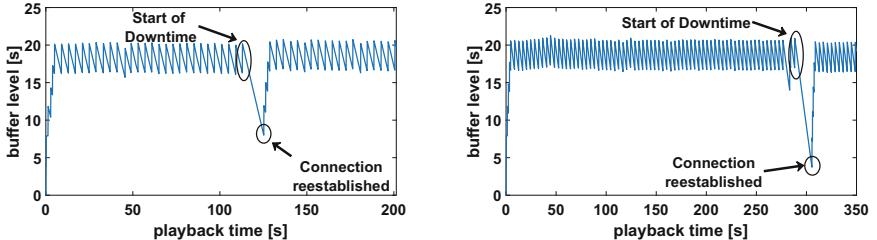
In *Scenario I* the migration is always performed - meaning the virtual machine is moved to the other server and the ffserv process continues its operation. Unfortunately, in 48% of the migration runs the client disconnects during the postoperation phase. If the cache is enabled, the playback continues until cache is depleted, as seen in Fig. 6. If no cache is enabled, the playback instantly terminates. It is required to restart the player to resume playback. Therefore, this scenario is only *semi-migrateable*, as a high chance of failure renders it inapplicable for production usage where the end-user should, in best case, not recognize the migration.

A very interesting behavior occurs when migrating in *Scenario II*. One might usually expect that under worse conditions the QoE would further decrease during the migration. This is not the case. Total migration takes longer especially due to the increased latency and is successfully completed as in *Scenario I*. The connection stays stable with the cache not filling for a short amount of time that correlates with the expected lengths of downtime and postoperation time. However, if no cache is enabled, the video stalls for a long period and then continues where it was suspended. This is a very surprising behavior. The only possible explanation is that the disconnect is not detected fast enough due to the already existing network delays. In this scenario the migration is *fully migrateable*.

## 6.3 Dynamic Adaptive Streaming over HTTP (DASH)

As a final scenario for video streaming, a client-based streaming application was chosen. On the server side the material was provided in multiple quality levels by

a web server and a playlist file was provided to the client. No further intelligence or optimization happened on the server side. The TAPAS player [17] (Tool for rApid Prototyping of Adaptive Streaming algorithms) was chosen on the client side. It is capable of adaptively streaming playback.



**Fig. 7.** Percentage of buffer fill level over time before, during and after migration using DASH in *Scenario I* (left) and *Scenario II* (right)

In *Scenario I* the migration is successfully performed. The server application survives the migration and the client successfully reconnects and continues streaming the video. During the time where no network operation is possible the clients cache level drops. With a preset cache level of 20s the migration is possible with fluid playback of the video stream as shown in Fig. 7. In the tested configuration after the first segment of the video the player constantly operates at the maximum quality possible even during the migration.

The experiment was repeated for *Scenario II*. The first obvious change was the far prolonged migration duration. The migration was again performed successfully without any negative effects on the server application. Again, the client reconnected successfully. This time the buffer level dropped farther, as seen in Fig. 7. Also, the application adapted due to the low buffer level, requesting the first segment after the reestablishment of the link in a lower quality level.

The migrations for the client side streaming application were always successful with no failures or total loss of connection, as observed for the server side applications. This renders this solution *completely migrateable* in both scenarios.

#### 6.4 Other Applications and Summary

We have also evaluated the migrateability of further applications. Migrating a simple file server in a m1.small flavor has been successful without any problems. Since this is a simpler version of the client side streaming this is little surprise.

We also have tested whether migrating a game server during a running game was possible. Therefore we ran a dedicated *Counter-Strike: Source* in a m1.medium flavor. The migration always succeeded but during the migration the game was unplayable. The preoperation phase caused major stuttering and

**Table 2.** Summary of the migratability of different applications. Where ✓ means the migration is possible and works reliable and ✗ means that it does not. ✗\* means that the migration is successful some times but fails at others or causes non tolerable impairments. Scenario 1 resembles intra compute center migrations while Scenario 2 resembles inter compute center migrations.

Application	Scenario 1	Scenario 2
Download	✓	✓
Server side streaming + content	✗	✗
Server side streaming w/o content	✗*	✓
Client side streaming	✓	✓
Video gaming	✗*	✗

lags. The downtime caused a longer lag while during the postoperation phase the AI players continued to operate while the human players were still waiting to reconnect. Thus we designate this scenario as not migrate-able.

An overview of the migrate-ability can be found in Table 2.

## 7 Conclusion

Due to the increasing popularity of cloud services, resource management with respect to the users' perceived quality, as well as in terms of energy efficiency and cost, is becoming more and more important in a cloud infrastructure. A fundamental part within this resource management process in the cloud is the migration of cloud services. The decision whether the benefits of a migration justify the migration effort requires addition information and studies. This includes in particular the influence of the migration on the actual service performance, the duration of the migration, and influences of different network parameters on these factors.

In this paper an overview of the related work has been given and the background is presented. A testbed to measure migrations has been described. The effect of the network parameters throughput and drop rate has been analyzed, showing that both parameters have different effects on the three phases of a migration.

Next, the migration performance was analyzed from application perspective. To assess this performance multiple applications have been chosen and migrated inside the testbed while clients were connected and using the provided services. The assessed applications include server-based video streaming with and without content inside the virtual machine and adaptive streaming using DASH. While the applications with little server side intelligence had few problems with the migrations the more complex applications failed to migrate at all or suffered severe impairments to the usability. More precisely the server side video streaming with included content did not migrate at all while on the other hand, the DASH streaming migrated without problems.

Future work may deal with developing concepts to make migrations application-aware. Thus, allowing the migration process to adapt to the used application and improve the quality during migration resp. making migration possible at all. Additional measurements to decrease the granularity are to be taken and further parameters (e.g. I/O load) are to be evaluated. With enough additional data, a model for migration duration should be designed.

**Acknowledgment.** This work was partially supported by German Research Foundation (DFG) under Grant No. KO 3445/11-1. and the H2020 INPUT (Call H2020-ICT-2014-1, Grant No. 644672).

## References

1. Petley, C., Goasdouff, L.: Gartner Says Worldwide Public Cloud Services Market to Grow 18 Percent in 2017, February 2017
2. Fehling, C., Leymann, F., Ruehl, S.T., Rudek, M., Verclas, S.: Service migration patterns-decision support and best practices for the migration of existing service-based applications to cloud environments. In: 2013 IEEE 6th International Conference on Service-Oriented Computing and Applications (SOCA), pp. 9–16. IEEE (2013)
3. Piao, J.T., Yan, J.: A network-aware virtual machine placement and migration approach in cloud computing. In: 2010 9th International Conference on Grid and Cooperative Computing (GCC), pp. 87–92. IEEE (2010)
4. Mohammadi, E., Karimi, M., Heikalabad, S.R.: A novel virtual machine placement in cloud computing. Aust. J. Basic Appl. Sci. **5**(10), 1549–1555 (2011)
5. Hyser, C., McKee, B., Gardner, R., Watson, B.J.: Autonomic virtual machine placement in the data center. Technical report, Hewlett Packard Laboratories, HPL-2007-189, vol. 189 (2007)
6. Shrivastava, V., Zerfos, P., Lee, K.W., Jamjoom, H., Liu, Y.H., Banerjee, S.: Application-aware virtual machine migration in data centers. In: 2011 Proceedings of IEEE INFOCOM, pp. 66–70. IEEE (2011)
7. Huber, N.M.: Autonomic performance-aware resource management in dynamic IT service infrastructures. Ph.D. thesis, Karlsruhe, Karlsruher Institut für Technologie (KIT), Dissertation 2014 (2014)
8. Noorshams, Q., Busch, A., Rentschler, A., Bruhn, D., Kounev, S., Tuma, P., Reussner, R.: Automated modeling of I/O performance and interference effects in virtualized storage systems. In: 2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 88–93. IEEE (2014)
9. Vaupel, R., Noorshams, Q., Kounev, S., Reussner, R.: Using queuing models for large system migration scenarios – an industrial case study with IBM system z. In: Balsamo, M.S., Knottenbelt, W.J., Marin, A. (eds.) EPEW 2013. LNCS, vol. 8168, pp. 263–275. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-40725-3\\_20](https://doi.org/10.1007/978-3-642-40725-3_20)
10. Akoush, S., Sohan, R., Rice, A., Moore, A.W., Hopper, A.: Predicting the performance of virtual machine migration. In: 2010 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 37–46. IEEE (2010)
11. Liu, H., Jin, H., Xu, C.Z., Liao, X.: Performance and energy modeling for live migration of virtual machines. Clust. Comput. **16**(2), 249–264 (2013)

12. Sallam, A., Li, K.: A multi-objective virtual machine migration policy in cloud systems. *Comput. J.* **57**(2), 195–204 (2014)
13. Zheng, J., Ng, T.E., Sripanidkulchai, K., Liu, Z.: Pacer: a progress management system for live virtual machine migration in cloud computing. *IEEE Trans. Netw. Serv. Manag.* **10**(4), 369–382 (2013)
14. Kapil, D., Pilli, E.S., Joshi, R.C.: Live virtual machine migration techniques: survey and research challenges. In: 2013 IEEE 3rd International Advance Computing Conference (IACC), pp. 963–969. IEEE (2013)
15. Ahmad, R.W., Gani, A., Hamid, S.H.A., Shiraz, M., Xia, F., Madani, S.A.: Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues. *J. Supercomput.* **71**(7), 2473–2515 (2015)
16. Mathis, M., Mahdavi, J., Floyd, S., Romanow, A.: TCP Selective Acknowledgment Options. RFC 2018 (Proposed Standard), October 1996
17. De Cicco, L., Calderalo, V., Palmisano, V., Mascolo, S.: TAPAS: a tool for rApid prototyping of adaptive streaming algorithms. In: Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming, pp. 1–6. ACM (2014)



# A Cloud Service Enhanced Method Supporting Context-Aware Applications

Zifan Liu<sup>1(✉)</sup>, Qing Cai<sup>2</sup>, Song Wang<sup>3</sup>, Xiaolong Xu<sup>2</sup>, Wanchun Dou<sup>1</sup>, and Shui Yu<sup>4</sup>

<sup>1</sup> State Key Laboratory for Novel Software Technology, Nanjing University,  
Nanjing, China

[nju.liuzf@gmail.com](mailto:nju.liuzf@gmail.com), [douwc@nju.edu.cn](mailto:douwc@nju.edu.cn)

<sup>2</sup> School of Computer and Software,  
Nanjing University of Information Science and Technology, Nanjing, China  
[1392134662@qq.com](mailto:1392134662@qq.com), [njuxlxu@gmail.com](mailto:njuxlxu@gmail.com)

<sup>3</sup> State Grid Anhui Electric Power Company, Hefei, China  
[wsong09@163.com](mailto:wsong09@163.com)

<sup>4</sup> School of Information Technology, Deakin University, Melbourne, Australia  
[shui.yu@deakin.edu.au](mailto:shui.yu@deakin.edu.au)

**Abstract.** Mobile cloud computing is emerging as a powerful platform for running demanding applications migrated from mobile devices to a remote cloud. For some real-time or urgent deadline-constrained applications, the migration process generates intolerable transmission latency. Cloudlets co-located with Access Points (APs) are considered as an efficient way to reduce such transmission latency. However, it is still a challenge to manage the cloudlets that have been deployed for fixed context-aware applications to achieve cost savings. In view of this challenge, a cloud service enhanced method supporting context-aware applications is proposed in this paper. Specifically, a cloudlet management principle is designed to provide a reference for cloudlet status judgment. Then a relevant cloud service enhanced method is proposed to decide which active cloudlets should be shut down. Finally, the experimental and analytical results demonstrate the validity of our proposed method.

**Keywords:** Cloud service · Context-aware applications  
Mobile cloud computing · Cloudlet · Cost savings

## 1 Introduction

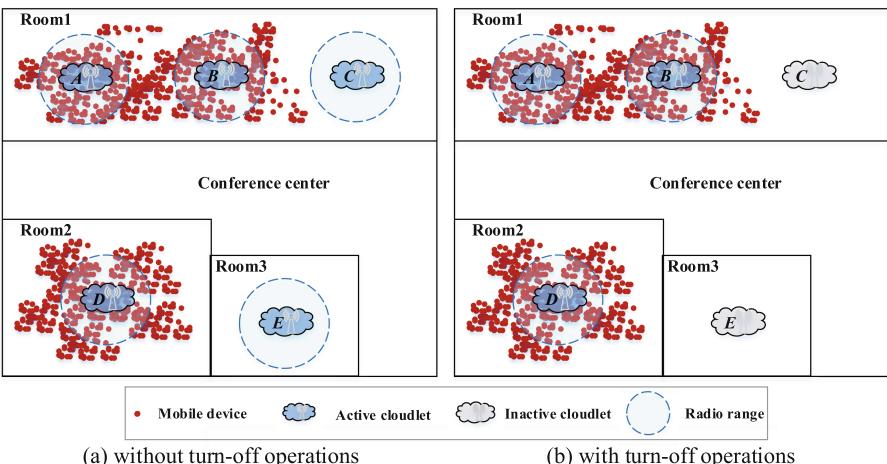
With increasing resource requirements of mobile applications in computation, communication and storage, mobile cloud computing is emerging as an effective way to realize on-demand resource provisioning for mobile applications [1, 2]. Mobile cloud computing provides abundant storage resources and high computing capability for migrated demanding applications, but at the same time, such application migration generates long transmission latency [1]. And the latency

is intolerable for users in some mobile applications, such as interactive gaming, speech recognition, video playback [3, 4], to name a few.

As an efficient and effective cloud deployment paradigm, cloudlets are deployed around mobile devices to reduce the transmission latency for mobile applications [5, 6]. Generally, cloudlets are resource-rich and self-managed which can be organized by harnessing the personal idle servers of other individual users or directly provisioned by the network operators (NOs) [1]. Mobile devices can offload their mobile applications to nearby cloudlets rather than remote clouds for processing. As a result, the physical proximity between mobile users and cloudlets is beneficial to transmission time reduction for workload offloading.

The cloudlets are in active mode when they are placed to enhance the local cloud services supporting context-aware applications. These active cloudlets consume overwhelming power which leads to high running cost. When there are no mobile devices within cloudlet coverage, the cloudlets in active mode are unnecessary to run. Such cloudlets should be shut down to achieve cost savings for energy consumption.

Here, an example of cloudlet management is presented to illustrate the problem investigated in this paper. There are a number of mobile devices with fixed context-aware applications, and 5 cloudlets, i.e.,  $A, B, C, D$ , and  $E$ , with APs distributed in 3 rooms (i.e., Room1, Room2, and Room3) of a conference center, as shown in Fig. 1. In Fig. 1, cloudlets  $A, B, C$  are placed in Room1, cloudlets  $D$  and cloud  $E$  are placed in Room2 and Room3, respectively. In Fig. 1(a), there are no mobile devices within the coverage of cloudlets  $C$  and  $E$ , but they consume a certain amount of energy. In such circumstance, cloudlets  $C$  and  $E$  should be shut down to achieve cost savings as illustrated in Fig. 1(b).



**Fig. 1.** An example of cloudlet management

However, current researches mainly focus on cloudlet placement to reduce energy consumption and network delay [7, 8]. Based on these observations, it is still a challenge to manage the cloudlets that have already been deployed for fixed context-aware applications to achieve cost savings.

In view of this challenge, a cloud service enhanced method supporting context-aware applications is proposed in this paper. Specifically, a cloudlet management principle is designed to provide a reference for cloudlet status judgments. Then a cloud service enhanced method is proposed to decide which active cloudlets should be shut down. Finally, the experimental results demonstrate that the proposed method is both effective and efficient.

The rest of the paper is organized as follows. A cloudlet management principle is presented in Sect. 2. Section 3 elaborates a cloud service enhanced method supporting context-aware applications. Experimental evaluations are conducted in Sect. 4 to demonstrate the validity of our method. Some related work is described in Sect. 5. Section 6 concludes the paper and gives an outlook on possible future work.

## 2 Cloudlet Management Principle

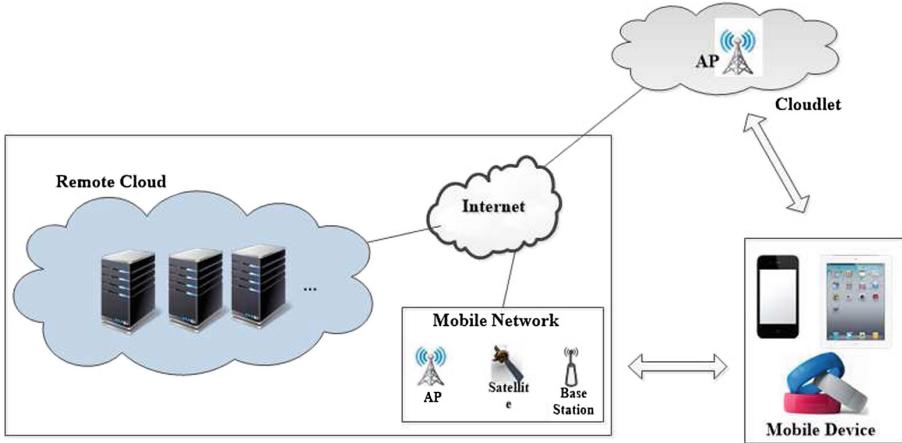
In this section, formalized concepts are given to facilitate our further discussion for cloud service enhancement supporting context-aware applications. To simplify the discussion, key terms used in our cloudlet management principle are summarized in Table 1.

**Table 1.** Key terms and descriptions

Terms	Description
$S$	The set of the points in device activity area
$DA$	The set of device activity area, $DA = \{da_1, da_2, \dots, da_N\}$
$da_n$	The $n$ -th device activity area
$MD$	The set of mobile devices, $MD = \{md_1, md_2, \dots, md_M\}$
$md_n$	The set of mobile devices in $da_n$ , $md_n = \{md_{n,1}, md_{n,2}, \dots, md_{n,Z}\}$
$mp_m$	The position $md_m$ , $mp_m = (mpx_m, mpy_m)$
$cl_{n,i}$	The $i$ -th cloudlet in $da_n$
$cpx_{n,i}$	The central position of $cl_{n,i}$ , $cpx_{n,i} = (cpx_{n,i}, cpy_{n,i})$
$dc_{n,i}$	The device collection of $cl_{n,i}$
$r_{n,i}$	The coverage radius for $cl_{n,i}$
$IC_{n,i}$	The indoor cloudlet coverage collection of $cl_{n,i}$
$\rho$	The density threshold for cloudlet placement judgment

To enhance cloud service supporting fixed context-aware applications, a 3-tier mobile cloud infrastructure with cloudlet is introduced, as shown in Fig. 2.

In this infrastructure, the cloudlets are co-located with APs, the remote cloud provides data storage and computing service, and the mobile devices are clients that can access the mobile cloud service through wireless networks. Compared to the traditional client-server communication model [1] without cloudlet, it is more powerful to reduce access latency by leveraging such 3-tier infrastructure. Via the cloudlets placed nearby, mobile devices can get direct cloud computing resources through AP to access cloudlets in the network.



**Fig. 2.** Mobile cloud infrastructure with cloudlet

Generally, there are various shapes of cloudlet management areas in practice. As other shaped areas can be assembled by multiple rectangles of different sizes, the cloudlet management area is defined by a rectangle in this paper.

**Definition 1 (Cloudlet Management Area).** *Cloudlet management area is defined by the  $x-y$  plane with definite ranges of  $x$ -axis and  $y$ -axis, denoted as  $S = \{(x, y) | 0 \leq x \leq X, 0 \leq y \leq Y\}$ .*

Within the cloudlet management area, there are a number of device activity areas, denoted as  $DA = \{da_1, da_2, da_N\}$  ( $DA \subseteq S$ ), where  $N$  is the number of device activity areas in  $S$ . Each device activity area has the  $x-y$  plane to show the range.

**Definition 2 (Divisional Device Activity Area).** *For the  $n$ -th ( $1 \leq n \leq N$ ) device activity area  $da_n$ , it is defined by the  $x-y$  plane with definite ranges of  $x$ -axis and  $y$ -axis, it is denoted as  $da_n = \{(dax_n, day_n) | 0 \leq dax_n \leq X_n, 0 \leq day_n \leq Y_n\}$ .*

Within the cloudlet management area, there are a number of mobile devices distributed randomly, denoted as  $MD = md_1, md_2, \dots, md_M$ , where  $M$  is the number of devices in  $S$ . Each mobile device has a position to show their locations.

**Definition 3 (Mobile Device Position).** For the  $m$ -th ( $1 \leq m \leq M$ ) mobile device  $md_m$ , the device is located at some positions that is a 2-tuple in  $S$ , denoted as  $mp_m = (mpx_m, mpy_m)$ , where  $mpx_m$  and  $mpy_m$  are the  $x$ -axis value and the  $y$ -axis value of  $mp_m$ , respectively.

Each cloudlet is co-located with an AP. Let  $cl_n$  be a collection of cloudlets located within  $da_n$  and  $z_n$  ( $z_n \geq 1$ ) be the total number of cloudlets in  $cl_n$ . Then the total number  $K$  of cloudlets in  $S$  can be calculated by

$$K = \sum_{n=1}^N z_n \quad (1)$$

Each deployed cloudlet has a central position, which is beneficial to confirm the covered mobile devices.

**Definition 4 (Cloudlet Central Position).** For the  $i$ -th ( $1 \leq i \leq z_n$ ) cloudlet  $cl_{n,i}$  in the  $n$ -th device activity area  $da_n$ , the relevant central position is location where AP is placed, which is a 2-tuple in  $dan$ , denoted as  $cp_{n,i} = (cpx_{n,i}, cpy_{n,i})$ , where  $cpx_{n,i}$  is the  $x$ -axis value and  $cpx_{n,i}$  is the  $y$ -axis value of  $cl_{n,i}$ .

The cloudlet can enhance the cloud service for users within cloudlet coverage area. In order to identify whether the cloudlets need to turn off, it is important to detect the device collection of cloudlet central positions.

**Definition 5 (Cloudlet Coverage Collection).** For the  $i$ -th cloudlet  $cl_{n,i}$  in  $da_n$ , the corresponding device collection is defined by  $dc_{n,i} = \{md_m | dis(mp_m, cp_{n,i}) \leq r_{n,i}, 1 \leq m \leq M\}$ , where  $dis(mp_m, cp_{n,i})$  is calculated by

$$dis(mp_m, cp_{n,i}) = \sqrt{(mpx_m - cpx_{n,i})^2 + (mpy_m - cpy_{n,i})^2} \quad (2)$$

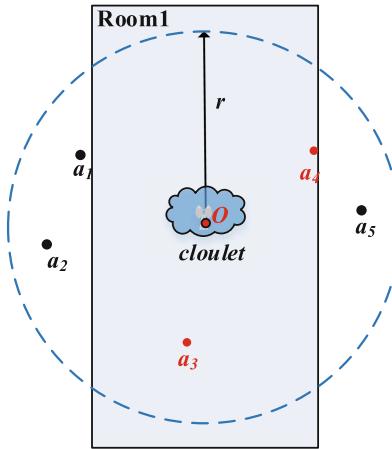
Some mobile devices may be covered simultaneously by multiple cloudlets in different areas. In the circumstances, the real device collection of a cloudlet central position should subtract the number of devices outdoor. Suppose there are a set of mobile devices in  $n$ -th device activity area  $da_n$ , denoted as  $MD_n = \{md_{n,1}, md_{n,2}, \dots, md_{n,Z}\}$ , where  $Z$  is the number of devices in  $dan$ . Then an indoor cloudlet coverage collection is defined as follows.

**Definition 6. Indoor Cloudlet Coverage Collection** For the  $i$ -th cloudlet  $cl_{n,i}$  in  $da_n$ , the corresponding indoor cloudlet coverage collection, denoted as  $IC_{n,i}$ , is defined by

$$IC_{n,i} = dc_{n,i} \cap MD_n \quad (3)$$

Figure 3 gives an example of indoor cloudlet coverage collection of *cloudlet*, the radius of *cloudlet* is  $r$  placed in the center  $O$ . The devices  $a_1, a_2, a_3, a_4$  and  $a_5$  are covered by *cloudlet*, but only devices  $a_3$  and  $a_4$  are in Room1. So the indoor cloudlet coverage collection of *cloudlet* is  $\{a_3, a_4\}$ .

In this paper, the coverage density plays a key role in our process of cloudlet. To achieve the goal of cost savings for mobile users within  $S$ , a cloudlet management principle is presented as a measurement for cloudlet status management.



**Fig. 3.** An example of indoor cloudlet coverage collection of *cloudlet*

**Definition 7.** *Cloudlet Management Principle* If the  $i$ -th cloudlet  $cl_{n,i}$  with central position  $G(x, y)$  runs in active mode to serve mobile users within  $da_n$ , the number of indoor cloudlet coverage collection  $IC_{n,i}$  of  $cl_{n,i}$  should satisfy the condition that  $|IC_{n,i}| \geq \rho$ , where  $\rho$  is a density threshold for cloudlet status management judgment.

### 3 Cloud Service Enhanced Method Supporting Context-Aware Applications

In this section, a cloud service enhancement method supporting context-aware applications is presented for the mobile cloudlet placement. This method consists of the three steps specified in detail by Fig. 4.

- Step1:** Mobile device positions identification. The positions of mobile devices are identified by the mobile location technology for further cloudlet management in the following steps.
- Step2:** Device collection generation. The device collection of each cloudlet is derived from the mobile device positions gained by Step1. Depending on device area constraints, devices outside will be removed from corresponding collections.
- Step3:** Cloudlet management. In this step, the device collections will be estimated based on the cloudlet threshold. A greedy algorithm is utilized for selecting the least necessary cloudlets. All the selected cloudlets are turned on while others are shut down.

**Fig. 4.** Specifications of cloud service enhanced method supporting context-aware applications

### 3.1 Step1: Mobile Device Positions Identification

In order to present our cloud service enhanced method, it is essential to locate the mobile devices for analysis. Currently, Mobile Location Based Service (MLBS) is a pretty proven technology, which can accurately locate the mobile devices via a mobile terminal or a personal geographical through wireless network [9, 10]. According to the basic principles of mobile location, MLBS can be broadly divided into two categories: network-based location technology and terminal-based location technology [11]. Besides, mobile devices also can be located via combining these two location technologies.

Here, considering the Internet surfers in a cloud service enhanced network, network-based location technology is more appropriate to locate the mobile devices. Even if the locations of mobile devices change, it can be accurately located in no time. Taking advantage of such technology, a set of device positions are identified for further cloudlet management.

### 3.2 Step2: Device Collection Generation

The cloudlets are dispersedly deployed all over the device area and they will affect the largest number of mobile devices if they are all active. Yet not all the cloudlets can be turned on according to the cost saving consideration. Besides, in real world, the mobile devices usually scatter in an uneven distribution, thus it is appropriate that we select several cloudlets to cover considerable mobile devices. On this purpose, the collection of mobile devices that each cloudlets covers is identified.

We consider the device positions  $MP = \{mp_1, mp_2, \dots, mp_N\}$  gained by Step1 and the coverage radius  $r$  of every cloudlet. For each cloudlet  $cl_{n,i}$ , all the mobile devices, the distances of which and  $cl_{n,i}$  are less than  $r$ , are added to corresponding device collection  $dc_i$ . Additionally, as  $cl_{n,i}$  is in the device area  $da_n$ , it cannot affect the devices outside of this area, thus all the mobile devices outside are deleted from  $dc_{n,i}$ .

Algorithm 1 specifies the generation process of device collections. For each collection, the devices outside the corresponding device area are deleted.

### 3.3 Step3: Cloudlet Management

Take advantage of Step2, the cloudlets which should be active are determined in this step. The device collections gained by Step2 differ in size and it is a waste to turn on a cloudlet which covers only few mobile devices. Therefore, a cloudlet management principle  $\rho$ , defined in Definition 5, is presented to filter the device collections. Concretely,  $\rho$  represents the least number of mobile devices that an active cloudlet should cover. All cloudlets that dissatisfy such threshold are shut down to reduce energy cost.

Furthermore, the coverage area of these cloudlets may overlap partly. Consider the case that numerous mobile devices gather in the overlapped area of two cloudlets and few in separate areas. In view of the cloudlet threshold, both two

**Algorithm 1.** Device Collection Generation( $MD, CL$ )

---

**Require:** A set of mobile device  $MD$  and a set of cloudlet  $CL$ .
**Ensure:** A set of device collections.

```

1: for  $i = 1 \rightarrow N$  do
2:   for  $j = 1 \rightarrow cl_n.size$  do
3:     if  $dis(mp_i, cl_{i,j}) < r_{i,j}$  then
4:       add  $md_i$  to  $dci_{i,j}$ 
5:     end if
6:   end for
7: end for
8: for  $i = 1 \rightarrow N$  do
9:   for  $j = 1 \rightarrow cl_n.size$  do
10:     $ic_{i,j} \leftarrow dci_{i,j}$ 
11:    for  $mp_k$  in  $dci_{i,j}$  do
12:      if  $mp_k$  is not in  $da_i$  then
13:        Delete  $mp_k$  from  $ic_{i,j}$ 
14:      end if
15:    end for
16:  end for
17: end for
```

---

cloudlets should be turned on, which generates a waste compared with turning on only one cloudlet of them. In order to overcome such problem, a greedy algorithm is employed in our method. First, the largest device collection, denoted as  $dc_i$ , is chosen and  $cl_i$  is added in result collection, denoted as  $clo$ . Second, for all the mobile devices in  $dc_i$ , delete them from every device collection. Repeat the two steps until sizes of all device collections are smaller than  $\rho$ . Then the cloudlets in result collection are turned on, while remaining cloudlets are shut down.

Algorithm 2 specifies management process of all cloudlets in device management area. The set of device collections is generated by Step1. All the cloudlets in  $clo$  are turned on while others are shut down.

## 4 Experimental Evaluation

In this section, HANA in-memory database is employed to evaluate the performance of our proposed cloud service enhanced method supporting context-aware applications.

### 4.1 Experiment Settings

In our experiments, two nodes are engaged to create a HANA cloud, including a master and a slave, and the configuration of the hardware and the software is specified in Table 2.

To simplify the experimental evaluation on our method, a case study is presented. The shape of device management area is a square with sides of 140 m.

**Algorithm 2.** Cloudlet management( $IC$ )

---

**Require:** A set of device collections  $IC$ .

**Ensure:** The cloudlet management policy.

```

1:  $clo \leftarrow \emptyset$ 
2: for  $i = 1 \rightarrow N$  do
3:   while true do
4:      $ic$  is the largest device collection in  $da_i$ 
5:     if  $|ic| > \rho$  then
6:       add  $ic$  to  $clo$ 
7:       for each  $ic_{i,k}$  in  $da_i$  do
8:          $ic_{i,k} \leftarrow ic_{i,k} - ic$ 
9:       end for
10:    else
11:      break
12:    end if
13:   end while
14: end for
15: for  $ic_{i,j}$  in  $IC$  do
16:   if  $ic_{i,j}$  is in  $clo$  then
17:     turn on  $cl_{i,j}$ 
18:   end if
19: end for
```

---

**Table 2.** The experiment context

	Client	HANA Cloud
Hardware	Lenovo Thinkpad T430 machine with Intel i5-3210M 2.50 GHz processor, 4 GB RAM and 250 GB Hard Disk	Master/slave: HP Z800 Workstation Intel(R) Multi-Core X5690 Xeon(R), 3.47 GHz/12M Cache, 6cores, 2 CPUs, 128 GB (8 × 8 GB + 4 × 16 GB) DDR3 1066 MHz ECC Reg RAM; 1 disk on the master and 2 disks on the slave: 2TB,7.2K RPM SATA Hard Drive
Software	Windows 7 Professional 64bit OS and HANA Studio	SUSE Enterprise Linux Server 11 SP3 and SAP HANA Platform SP07

There are five separate device activity areas, denoted as area  $a, b, c, d, e$ , in the device management area. In each device activity area, one or more cloudlets have been deployed and most space of those areas is covered by at least one cloudlet. Over the whole device management area, a set of mobile device positions is randomly generated. In practice, the cloudlet management principle  $\rho$  is set to 15 and the coverage radius  $r$  is set to 25 m. The detailed experimental parameters are specified in Table 3.

**Table 3.** The experiment context

Parameter item	Domain
The maximum $x$ -axis value $X$	140 m
The maximum $y$ -axis value $Y$	140 m
Cloudlet management principle $\rho$	15
The radio range $r$ of all mobile cloudlets	25 m
The number of mobile devices	200
The maximum $x$ -axis value $X$	140 m

## 4.2 Performance Evaluation

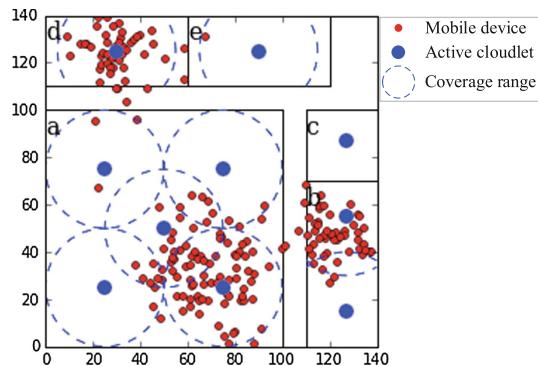
In this section, performance evaluations are presented to discuss the coverage number of mobile devices by using our method compared with this value without cloudlet management. For the device activity areas without cloudlet management, these cloudlets cannot be shut down. As numerous mobile devices are distributed in the device management area, the cloudlet covered mobile devices under the above two situations, i.e., with cloudlet management by our method and without cloudlet management are presented.

In Fig. 5, the distribution of mobile devices is shown without cloudlet management. Most devices gather in area  $a$ ,  $b$ , and  $d$ . Concretely, the right bottom part of area  $a$ , the upper part of area  $b$  and the center part of area  $d$  are the dense region of mobile devices. In our experiments, the walls of each activity area are radiation resistant and the cloudlet cannot affect the mobile devices that outside of corresponding area. The experimental results are illustrated by Fig. 6.

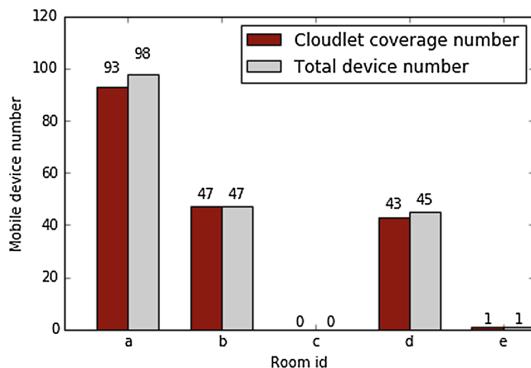
In Fig. 6, the cloudlet coverage number in each activity area is compared with total device number in such area. From this comparison, we can find that nearly all mobile devices are covered by cloudlets. Under this situation, all cloudlets are active that generates a certain amount of running costs.

Then our method is conducted on the same conditions. The management results of our method after cloudlet management are presented in Fig. 7. Compared with Fig. 5, only 4 of the 10 cloudlets are active. Furthermore, the cloudlet coverage number, as shown by Fig. 8, is 175 out of total 200, which is quite close to the number, 184 out of 200, before leveraging our method.

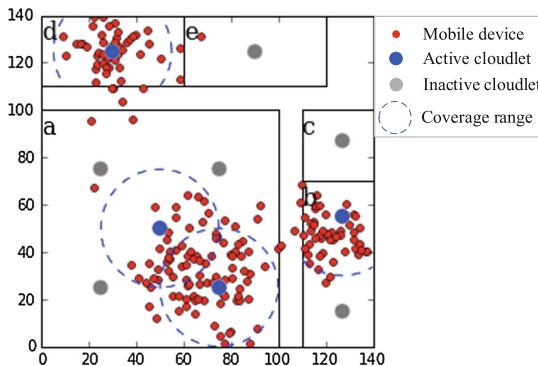
From the experimental evaluation, we can find after cloudlet management by our method, the cloudlet coverage value is similar to this value without cloudlet management. But our method can greatly reduce the energy consumption and the running costs of cloudlets, since only 4 of total 10 cloudlets running after cloudlet management.



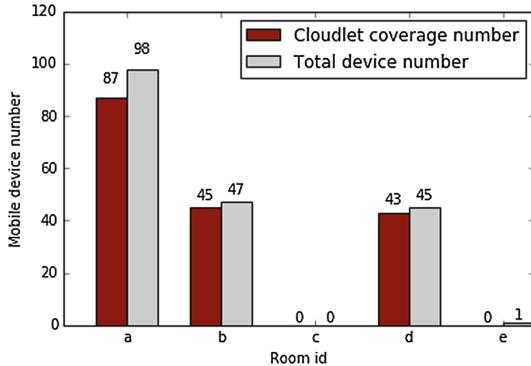
**Fig. 5.** Mobile device distribution and cloudlets status before using our method



**Fig. 6.** Cloudlet coverage numbers in each device activity area before using our method



**Fig. 7.** Mobile device distribution and cloudlets status after using our method



**Fig. 8.** Cloudlet coverage numbers in each device activity area after using our method

## 5 Related Work and Comparison Analysis

Currently, many researches of mobile cloud computing have been proposed to improve the computing capacity of mobile devices [1, 12]. However, the clouds are geographically far away from mobile users, in the result of which a huge latency generates during the workload offloading between mobile devices and remote clouds. Thus cloudlet is applied close to users to provide compute capability and data storage, which can greatly decrease the response time. They have been fully investigated in [13–17], to name a few.

In [13], the author proposed a network architecture, which is cost-effective via combining localized and distributed mini-clouds and fast-deployable wireless mesh networks. Xia *et al.* [14] devised an efficient online algorithm to solve an online location-aware offloading problem in a two-tiered mobile cloud computing environment, which consists of a local cloudlet and remote clouds. In [16], a framework named PEFC (Performance Enhancement Framework of Cloudlet) was proposed to enhance the finite resource cloudlet performance by increasing cloudlet resources. Artail *et al.* [17] formulated a more ubiquitous solution to decrease high network latency of remote cloud services, which relies on a network of cloudlets distributed within a geographic area. Despite the increasing momentum of cloudlet research, the cost of cloudlets has largely been overlooked.

Current researches mainly focus on cloudlet placement to reduce energy consumption and network delay. Wu and Ying [18] designed a novel virtual currency tailored for the cloudlet-based multi-lateral resource exchange framework, which made the resource exchange efficiently. Ravi and Peddoju [19] proposed a model to tackle the mobility and energy efficiency of mobile cloud computing. But they ignore the energy consumption of the redundant cloudlets. The best solution is to effectively manage the cloudlets usage. Thus we formulate a cloudlet management principle to save energy consumption of the cloudlets.

## 6 Conclusion and Future Work

In this paper, a cloud service enhanced method supporting context-aware applications is proposed in this paper. Specifically, a cloudlet management principle is designed to provide a reference for cloudlet status judgments. Then a cloud service enhanced method is proposed to decide which active cloudlets should be shut down. Finally, the experimental and analytical results demonstrate that the proposed method is both effective and efficient.

For future work, we plan to apply our cloud service enhanced method to real-world cloudlet platform. Besides, we will analyze energy consumption of the cloudlet management in different environment. And we intend to find a better energy-efficient method to manage cloudlets status.

**Acknowledgement.** This research is partially supported by the National Science Foundation of China under Grant No. 61672276, No. 61702277, the Key Research and Development Project of Jiangsu Province under Grant No. BE2015154 and BE2016120, and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University.

## References

1. Li, Y., Gao, W.: Code offload with least context migration in the mobile cloud. In: IEEE Conference on Computer Communications (INFOCOM), pp. 1876–1884 (2015)
2. Xu, Z., Liang, W., Xu, W., Jia, M., Guo, S.: Efficient algorithms for capacitated cloudlet placements. *IEEE Trans. Parallel Distrib. Syst.* **27**(10), 2866–2880 (2016)
3. Cohen, J.: Embedded speech recognition applications in mobile phones: status, trends, and challenges. In: IEEE International Conference on Acoustics, Speech and Signal (ICASSP), pp. 5352–5355 (2008)
4. Mohiuddin, K., Mohammad, A.R., Raja, A.S., Begum, S. F.: Mobile-CLOUD-mobile: is shifting of load intelligently possible when barriers encounter? In: IEEE International Conference on Information Science and Digital Content Technology (ICIDT), vol. 2, pp. 326–332 (2012)
5. Rimal, B.P., Van, D.P., Maier, M.: Cloudlet enhanced fiber-wireless access networks for mobile-edge computing. *IEEE Trans. Wirel. Commun.* **16**(6), 3601–3618 (2017)
6. Dinh, H.T., Lee, C., Niyato, D., Wang, P.: A survey of mobile cloud computing: architecture, applications, and approaches. *Wirel. Commun. Mob. Comput.* **13**(18), 1587–1611 (2011)
7. Xu, Z., Liang, W., Xu, W., Jia, M., Guo, S.: Capacitated cloudlet placements in wireless metropolitan area networks. In: IEEE Conference on Local Computer Networks (LCN), pp. 570–578 (2015)
8. Jia, M., Cao, J., Liang, W.: Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. *IEEE Trans. Cloud Comput.* **5**, 2168–7161 (2015)
9. Wang, H., Wang, Z., Shen, G., Li, F., Han, S., Zhao, F.: WheelLoc: enabling continuous location service on mobile phone for outdoor scenarios. In: IEEE Conference on Computer Communications (INFOCOM), pp. 2733–2741 (2013)

10. Constandache, I., Gaonkar, S., Sayler, M., Choudhury, R.R., Cox, L.: EnLoc: energy-efficient localization for mobile phones. In: IEEE Conference on Computer Communications (INFOCOM), pp. 2716–2720 (2009)
11. Lee, H.J., Wicke, M., Kusy, B., Guibas, L.: Localization of mobile users using trajectory matching. In: Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments, pp. 123–128 (2008)
12. Chen, X.: Decentralized computation offloading game for mobile cloud computing. *IEEE Trans. Parallel Distrib. Syst.* **26**(4), 974–983 (2015)
13. Khan, K.A., Wang, Q., Grecos, C., Luo, C., Wang, X.: MeshCloud: integrated cloudlet and wireless mesh network for real-time applications. In: IEEE Conference on Electronics, Circuits, and Systems (ICECS), pp. 317–320 (2013)
14. Xia, Q., Liang, W., Xu, Z., Zhou, B.: Online algorithms for location-aware task offloading in two-tiered mobile cloud environments. In: IEEE/ACM 7th International Conference on Utility and Cloud Computing, pp. 109–116 (2014)
15. Bahtovski, A., Gusev, M.: Multilingual cloudlet-based dictionary. In: IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 380–385 (2014)
16. Whaiduzzaman, M., Gani, A., Naveed, A.: PEFC: performance enhancement framework for cloudlet in mobile cloud computing. In: IEEE International Symposium on Robotics and Manufacturing Automation (ROMA), pp. 224–229 (2014)
17. Artail, A., Frenn, K., Safa, H., Artail, H.: A framework of mobile cloudlet centers based on the use of mobile devices as cloudlets. In: IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 777–784 (2015)
18. Wu, Y., Ying, L.: A cloudlet-based multi-lateral resource exchange framework for mobile users. In: Proceedings of IEEE Conference on Computer Communications (INFOCOM), pp. 927–935 (2015)
19. Ravi, A., Peddoju, S.K.: Mobility managed energy efficient Android mobile devices using cloudlet. In: IEEE Conference on Students' Technology Symposium (Tech-Sym), pp. 402–407 (2014)



# Application of 3D Delaunay Triangulation in Fingerprint Authentication System

Wencheng Yang<sup>1()</sup>, Guanglou Zheng<sup>1</sup>, Ahmed Ibrahim<sup>1</sup>, Junaid Chaudhry<sup>1</sup>, Song Wang<sup>2</sup>, Jiankun Hu<sup>3</sup>, and Craig Valli<sup>1</sup>

<sup>1</sup> Security Research Institute, School of Science, Edith Cowan University, Perth, WA 6027, Australia

{w.yang, g.zheng, ahmed.ibrahim, j.chaudhry, c.valli}@ecu.edu.au

<sup>2</sup> School of Engineering and Mathematical Sciences, La Trobe University, Melbourne, VIC 3086, Australia

Song.Wang@latrobe.edu.au

<sup>3</sup> School of Engineering and Information Technology, University of New South Wales at Canberra, Canberra, ACT 2600, Australia

J.Hu@adfa.edu.au

**Abstract.** Biometric security has found many applications in Internet of Things (IoT) security. Many mobile devices including smart phones have supplied fingerprint authentication function. However, the authentication performance in such restricted environment has been downgraded significantly. A number of methods based on Delaunay triangulation have been proposed for minutiae-based fingerprint matching, due to some favorable properties of the Delaunay triangulation under image distortion. However, all existing methods are based on 2D pattern, of which each unit, a Delaunay triangle, can only provide limited discrimination ability and could cause low matching performance. In this paper, we propose a 3D Delaunay triangulation based fingerprint authentication system as an improvement to improve the authentication performance without adding extra sensor data. Each unit in a 3D Delaunay triangulation is a Delaunay tetrahedron, which can provide higher discrimination than a Delaunay triangle. From the experimental results it is observed that the 3D Delaunay triangulation based fingerprint authentication system outperforms the 2D based system in terms of matching performance by using same feature representation, e.g., edge. Furthermore, some issues in applying 3D Delaunay triangulation in fingerprint authentication, have been discussed and solved. To the best of our knowledge, this is the first work in literature that deploys 3D Delaunay triangulation in fingerprint authentication research.

**Keywords:** 3D Delaunay triangulation · Fingerprint authentication

## 1 Introduction

The applications of fingerprint authentication can be found in both civil and military facets, e.g., Internet of Things (IoT), border control and financial transactions. As compared to other biometrics, such as face, voice, palm, ECG etc. [1–4], fingerprint

based authentication systems occupy the most market share because of the stability and distinctiveness that fingerprint can provide. However, fingerprint matching is not an easy task due to the fingerprint uncertainty caused by distortion, rotation and translation during the fingerprint image acquisition process. In order to mitigate the negative influence of fingerprint uncertainty, Delaunay triangulation based local structures have been proposed and studied by many existing methods, e.g., [5–8], for some specific local and global features that Delaunay triangulation can provide. First, each minutia in the Delaunay triangulation keep stable structure with its neighbors, despite a certain degree of non-linear distortion happens, which means that it has a stable local structure. Second, spurious and missing minutiae only influence the local units that contain those minutiae.

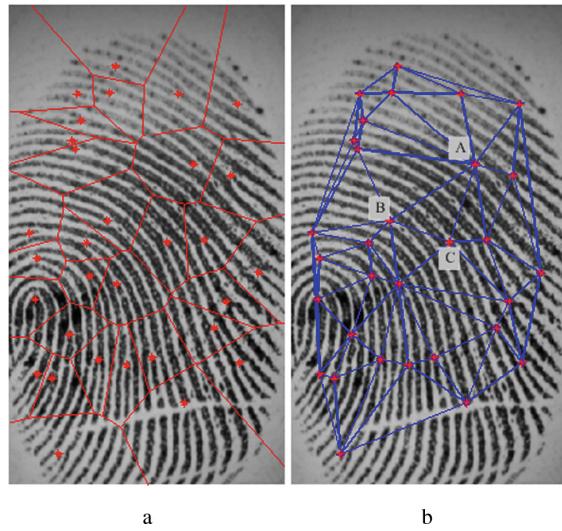
However, all existing methods in fingerprint authentication that include Delaunay triangulation are based on 2D pattern [9–14]. One drawback of using 2D Delaunay triangulation is that each unit, e.g., Delaunay triangle  $\Delta ABC$ , as shown in Fig. 1b, contains only a few features, three edges and three angles, which limit its distinctive capabilities and could lead to low system matching performance. Motivated by this, in this paper, we propose a 3D Delaunay triangulation based fingerprint authentication system. The main contribution of our work is two-fold: First, each unit in a 3D Delaunay triangulation (Fig. 2a) is a tetrahedron, e.g.,  $\blacktriangle ABCD$ , as shown in Fig. 2b. Instead of three edges and three angles, each tetrahedron consists of six edges and twelve angles, which can provide higher distinctiveness than a triangle. Second, issues such as data normalization and local structure registration during the process of applying 3D Delaunay triangulation in the fingerprint authentication are discussed and solved.

## 2 2D and 3D Delaunay Triangulation Construction

In this section, we first introduce the 2D Delaunay triangulation construction. Given a fingerprint image which contains a set of minutiae  $M = (m_1, m_2, m_3, \dots, m_N)$ , each minutia  $m_{i \in [1, N]}$  can be represented by a vector  $(x_i, y_i, \theta_i, t_i)$ , where  $(x_i, y_i)$  is the Cartesian coordinate of the minutiae location,  $\theta_i$  is the orientation of its associated ridge, and  $t_i$  is the minutia type. The generation of a 2D Delaunay triangulation only uses the coordinate  $(x_i, y_i)$  of each minutia and is based on the Voronoi tessellation which divides the whole fingerprint image into several smaller regions centering at each minutiae [15], as shown in Fig. 1a (red lines). By connecting the centers of every neighboring region, a 2D Delaunay triangulation is generated, as shown in Fig. 1b (blue lines). We have outlined the rationale for proposing the 3D Delaunay triangulation for the reason mentioned in the Introduction section. However, deployment of a 3D Delaunay triangulation in a fingerprint authentication system is not trivial, due to data normalization and local structure registration issues, thus need to be solved.

### 2.1 Data Normalization

During the generation of a 2D Delaunay triangulation, only two dimensions  $(x_i, y_i)$  of each minutia are needed, in contrast to three dimensions needed in a 3D Delaunay



**Fig. 1.** (a) Minutiae-based Voronoi tessellation. (b) The 2D Delaunay triangulation with Delaunay triangle  $\Delta ABC$  as an example. (Color figure online)

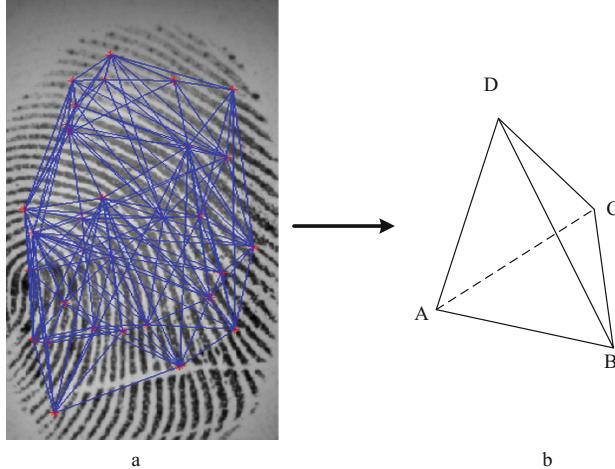
triangulation. In our application,  $\theta_i$  is treated as the third dimension. For any minutia  $m_{i \in [1, N]}$  extracted from a fingerprint image, its coordinate  $(x_i, y_i)$  are not on the same scale as the orientation  $\theta_i$ . Take minutia  $m_{i \in [1, N]}$  extracted by the commercial fingerprint recognition software Verifinger SDK [16] from fingerprint images of the public database FVC2002 DB2 [17] as an example. Its coordinate  $(x_i, y_i)$  is in the range of  $[0, 600]$  but its orientation  $\theta_i$  is in the range of  $[0, 2\pi]$ . To solve this issue, a normalization function is proposed as

$$N(\theta_i) = f_n(\theta_i, r_1[0, 2\pi], r_2[0, 600]) \quad (1)$$

where  $f_n(\cdot)$  is a normalization function that changes the value of  $\theta_i$  from the range of  $[0, 2\pi]$  to the range of  $[0, 600]$ ;  $r[x, y]$  represents the range between  $x$  and  $y$ . After the data normalization, a 3D Delaunay triangulation is generated, as shown in Fig. 2a.

## 2.2 Local Structure Registration

After the 3D Delaunay triangulation is constructed, one issue that needs to be solved before using the Delaunay tetrahedron for matching, is local structure registration. For example, there is a pair of corresponding Delaunay tetrahedrons,  $\blacktriangle ABCD$  and  $\blacktriangle A_1B_1C_1D_1$ , from the template and query images, respectively. In this scenario, the correct local structure registration is about precisely finding out the corresponding vertexes of  $A$ ,  $B$ ,  $C$  and  $D$  from  $\blacktriangle A_1B_1C_1D_1$ . To simplify the issue of local structure registration, we use an absolute geometric measurement to decide which vertex is the top vertex. Specially, areas of four surfaces of a tetrahedron are calculated and the vertex opposite the largest surface is chosen as the top vertex. We assume that  $D$  is



**Fig. 2.** (a) The 3D Delaunay triangulation. (b) A Delaunay tetrahedron  $\blacktriangle ABCD$  from the 3D Delaunay triangulation.

determined as the top vertex of  $\blacktriangle ABCD$ . Once the top vertex  $D$  is determined, the triangle  $\Delta ABC$  is defined as the base. We then search the vertex of the smallest angle from  $\Delta ABC$  and define it as the starting vertex  $A$  and the following vertexes as  $B$  and  $C$  in the anti-clockwise direction. By applying this measurement to the tetrahedron  $\blacktriangle A_1B_1C_1D_1$ , it is quite efficient to sort the vertexes  $A_1$ ,  $B_1$ ,  $C_1$  and  $D_1$ , which are corresponding to vertexes of  $A$ ,  $B$ ,  $C$  and  $D$ , respectively. Note that if two surfaces have the same largest size, the top vertex can be chosen from either one of the two vertexes.

### 3 3D Delaunay Triangulation Based Fingerprint Authentication

Delaunay tetrahedron based local structure extracted from the 3D Delaunay triangulation is employed for authentication in this paper. Given a template fingerprint image  $f^T$ , the whole fingerprint image is divided into  $N^T$  local structures  $\{L_i^T\}_{i=1}^{N^T}$ , where  $N^T$  is the number of minutiae in  $f^T$ , and each local structure  $L_i^T$  is composed by all the Delaunay tetrahedrons constituted by a centre minutiae and its  $K$  nearest neighbor minutiae in the local area. Assuming that the  $i^{\text{th}}$  local structure  $L_i^T$  contains  $N_{L_i^T}^T$  Delaunay tetrahedrons, and all vertexes of each tetrahedron are sorted similar as  $\blacktriangle ABCD$  by the absolute geometric measurement. In the experiments, we only use the edge length as features for matching, so that the  $i^{\text{th}}$  local structure  $L_i^T$  can be represented by

$$L_i^T = \{eAB_j, eBC_j, eCA_j, eDA_j, eDB_j, eDC_j\}_{j=1}^{N_{L_i^T}^T} \quad (2)$$

where  $eAB$  is the quantized edge length between the vertexes  $A$  and  $B$  using the same way in [14], and the quantization step size is  $q_e$ . Then the template fingerprint image  $f^T$  is expressed as a feature set  $V^T = \{L_i^T\}_{i=1}^{N^T}$ , which is stored in the database as a template in the enrolment stage. In the verification stage, given a query fingerprint image  $f^Q$ , we apply the same feature extraction and representation approach to it and obtain a feature set  $V^Q = \{L_j^Q\}_{j=1}^{N^Q}$ , where  $N^Q$  is the number of minutiae from  $f^Q$ . The matching score between  $L_i^T$  and  $L_j^Q$  depends on how many Delaunay tetrahedrons are matched. Two tetrahedrons are considered to be a match, if and only if all the six quantized edge lengths of two tetrahedrons are the same. Assuming the number of matched tetrahedron between  $L_i^T$  and  $L_j^Q$  are  $N_{ij}$ , the similarity score between them is calculated by

$$S(ij) = \frac{N_{ij}}{(N_{L_i}^T + N_{L_j}^Q)/2} \quad (3)$$

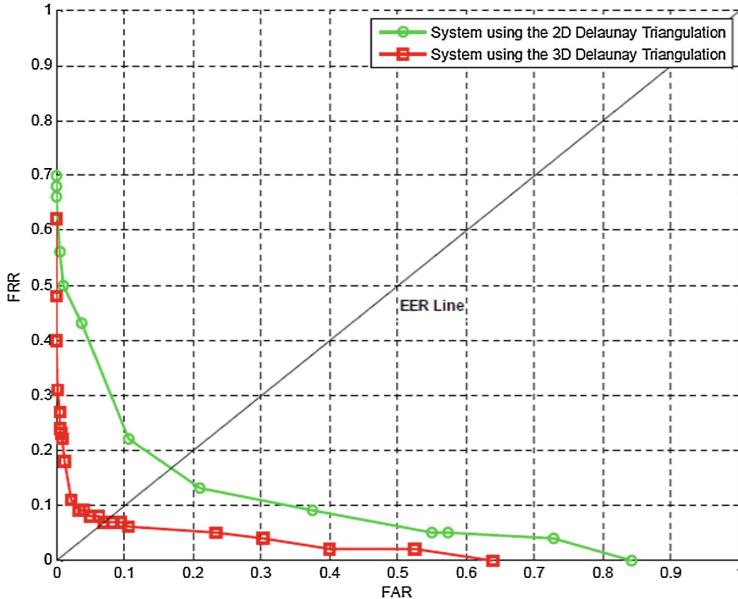
Each  $L_j^Q$  from the query, for  $1 \leq j \leq N^Q$ , has to be compared with each  $L_i^T$  from the template, for  $1 \leq i \leq N^T$ . By comparing all the local structures from template and query, a score matrix of size  $N^Q \times N^T$  is generated. If the largest value  $S_m$  from the score matrix is greater than a pre-defined threshold, the template and the query fingerprint images are considered a match.

## 4 Experimental Results and Analysis

We evaluated the proposed fingerprint authentication system over the public database FVC2002 DB2 [17], which includes 100 fingers with eight images per finger. The commercial fingerprint recognition software Verifinger SDK [16] is employed to extract minutiae from fingerprint images. The indicators, Equal Error Rate (EER), False Rejection Rate (FRR) and False Acceptance Rate (FAR), are adopted in our experiments to evaluate the system performance. Specifically, FRR is the rate of mistaking two fingerprint images from the same finger to be from two different fingers. The FAR is the rate of mistaking two different fingers to be from the same finger. The EER is the error rate when FRR and FAR are equal. We used the first image from each finger as the template and the second image from the same finger as the query image to calculate FRR, while we set the first image of each finger as template and the first image from all other different fingers as query image to calculate FAR.

To facilitate the comparison of the system matching performance, we only used the edge length of each unit as features to evaluate the matching performance in both 2D and 3D Delaunay triangulation based experiments. It is worth mentioning that other features, e.g., angle and minutia type, can be added to enhance the matching performance. Other parameters are set to be the same, for example, the quantization size  $q_e$  is set to be 20 and the value of  $K$  is 10 for both experiments. The performances of the system using the 2D and 3D Delaunay triangulation, respectively, are shown in Fig. 3, from which we can see that the performance of system using 3D Delaunay triangulation is EER = 7%, which is much better than the performance (EER = 6.37%) of system

using 2D Delaunay triangulation. This improved performance verifies that each unit, a Delaunay tetrahedron, from a 3D Delaunay triangulation can provide higher discrimination ability than the unit, a Delaunay triangle, from a 2D Delaunay triangulation.



**Fig. 3.** The performances of system using 2D Delaunay triangulation and 3D Delaunay triangulation, respectively.

## 5 Conclusion

In this paper, we applied a 3D Delaunay triangulation in fingerprint authentication rather than a 2D Delaunay triangulation utilized by existing work in the literature, so as to increase the discrimination ability of each unit in a Delaunay triangulation. Experimental results show that each unit from a 3D Delaunay triangulation is more distinctive than a single unit from a 2D Delaunay triangulation. Moreover, certain issues, e.g., data normalization and local structure registration, anticipated in the process of applying the 3D Delaunay triangulation in the fingerprint authentication system were discussed and solved. This is the first work applying 3D Delaunay triangulation in fingerprint authentication and we hope that this work would be a useful starting point for future research, e.g., in the area of cancelable biometrics [18–23].

**Acknowledgments.** This paper is supported by Defence Science and Technology Group (DST) of Australia through project CERA 221.

## References

1. Zheng, G., Fang, G., Shankaran, R., Orgun, M.A.: Encryption for implantable medical devices using modified one-time pads. *IEEE Access* **3**, 825–836 (2015)
2. Fang, G., Orgun, M.A., Shankaran, R., Dutkiewicz, E., Zheng, G.: Truthful channel sharing for self coexistence of overlapping medical body area networks. *PLoS ONE* **11**, e0148376 (2016)
3. Zheng, G., Shankaran, R., Orgun, M.A., Qiao, L., Saleem, K.: Ideas and challenges for securing wireless implantable medical devices: a review. *IEEE Sens. J.* **17**, 562–576 (2016)
4. Zheng, G., Fang, G., Shankaran, R., Orgun, M.A., Zhou, J., Qiao, L., Saleem, K.: Multiple ECG fiducial points-based random binary sequence generation for securing wireless body area networks. *IEEE J. Biomed. Health Inform.* **21**, 655–663 (2017)
5. Bebis, G., Deaconu, T., Georgopoulos, M.: Fingerprint identification using Delaunay triangulation. In: International Conference on Information Intelligence and Systems, pp. 452–459 (1999)
6. Liu, N., Yin, Y., Zhang, H.: A fingerprint matching algorithm based on Delaunay triangulation net. In: The 5th International Conference on Computer and Information Technology, pp. 591–595. IEEE (2005)
7. Wang, C., Gavrilova, M.L.: Delaunay triangulation algorithm for fingerprint matching. In: 3rd International Symposium on Voronoi Diagrams in Science and Engineering, pp. 208–216 (2006)
8. Yang, W., Hu, J., Wang, S., Stojmenovic, M.: An alignment-free fingerprint bio-cryptosystem based on modified Voronoi neighbor structures. *Pattern Recogn.* **47**, 1309–1320 (2014)
9. Junior, P., de Nazare-Junior, A., Menotti, D.: A complete system for fingerprint authentication using Delaunay triangulation. Re-conhecimento de Padroes, DECOM UFOP, pp. 1–7 (2010)
10. Yang, W., Hu, J., Stojmenovic, M.: NDTC: a novel topology-based fingerprint matching algorithm using N-layer Delaunay triangulation net check. In: 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 866–870. IEEE (2012)
11. Yang, W., Hu, J., Wang, S.: A Delaunay triangle-based fuzzy extractor for fingerprint authentication. In: 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 66–70. IEEE (2012)
12. Yang, W., Hu, J., Wang, S.: A Delaunay triangle group based fuzzy vault with cancellability. In: 2013 6th International Congress on Image and Signal Processing (CISP), pp. 1676–1681 (2013)
13. Yang, W., Hu, J., Wang, S., Yang, J.: Cancelable fingerprint templates with Delaunay triangle-based local structures. In: Wang, G., Ray, I., Feng, D., Rajarajan, M. (eds.) CSS 2013. LNCS, vol. 8300, pp. 81–91. Springer, Cham (2013). [https://doi.org/10.1007/978-3-319-03584-0\\_7](https://doi.org/10.1007/978-3-319-03584-0_7)
14. Yang, W., Hu, J., Wang, S., Chen, C.: Mutual dependency of features in multimodal biometric systems. *Electron. Lett.* **51**, 234–235 (2015)
15. Lee, D.-T., Schachter, B.J.: Two algorithms for constructing a Delaunay triangulation. *Int. J. Comput. Inf. Sci.* **9**, 219–242 (1980)
16. VeriFinger SDK: Neuro Technology. <<http://www.neurotechnology.com/verifinger.html>>
17. <http://bias.csr.unibo.it/fvc2002>
18. Wang, S., Hu, J.: A blind system identification approach to cancelable fingerprint templates. *Pattern Recogn.* **54**, 14–22 (2016)
19. Wang, S., Hu, J.: Design of alignment-free cancelable fingerprint templates via curtailed circular convolution. *Pattern Recogn.* **47**, 1321–1329 (2014)

20. Wang, S., Hu, J.: Blind channel estimation for single-input multiple-output OFDM systems: zero padding based or cyclic prefix based? *Wirel. Commun. Mob. Comput.* **13**, 204–210 (2013)
21. Wang, S., Hu, J.: Alignment-free cancellable fingerprint template design: a densely infinite-to-one mapping (DITOM) approach. *Pattern Recogn.* **45**, 4129–4137 (2012)
22. Wang, S., Deng, G., Hu, J.: A partial Hadamard transform approach to the design of cancelable fingerprint templates containing binary biometric representations. *Pattern Recogn.* **61**, 447–458 (2017)
23. Wang, S., Yang, W., Hu, J.: Design of alignment-free cancelable fingerprint templates with Zoned Minutia Pairs. *Pattern Recogn.* **66**, 295–301 (2017)



# The Public Verifiability of Public Key Encryption with Keyword Search

Binrui Zhu<sup>1</sup>, Jiameng Sun<sup>1</sup>, Jing Qin<sup>1,2(✉)</sup>, and Jixin Ma<sup>3</sup>

<sup>1</sup> School of Mathematics, Shandong University, Jinan, Shandong, China  
zhubinrui1509889@163.com, sunjiameng1991@163.com

<sup>2</sup> State Key Laboratory of Information Security,  
Institute of Information Engineering,  
Chinese Academy of Sciences, Beijing, China  
qinjing@sdu.edu.cn

<sup>3</sup> School of Computing and Mathematical Sciences,  
University of Greenwich, London, UK  
j.ma@greenwich.ac.uk

**Abstract.** Cloud computing has been widely recognized as the next big thing in this era. Users outsourced data to cloud server and cloud server provided service economic savings and various convenience for users. Public key encryption with keyword search (PEKS) which provides a solution for a third party user to search on remote data encrypted by data owner. Since the server may be dishonest, it can perform search operation on encrypted data and only return partial results. Therefore, it is necessary to verify the correctness and completeness of the search result. Existing PEKS schemes only support data receiver's private verification, however, in practice, we usually need anyone can verify the server's search result. In this paper, we propose a PEKS with public verifiability scheme, which can achieve the security of ciphertext indistinguishability, trapdoor indistinguishability, keyword guessing attack and public verifiability. Comparing previous PEKS schemes, our scheme is public verifiability, while keeping the encrypted data security in cloud server and search operation privately over the encrypted data.

**Keywords:** Cloud computing · PEKS · Public verifiability  
Indistinguishability

## 1 Introduction

With the advent of the cloud ear, more and more users would like to store their data to the cloud server. By moving data to the cloud server, it provides both economical saving and various convenience for users. Despite having these benefits, security is still considered as the major barriers for the user and enterprise. Cloud server may be honest but curious, in order to ensure the security, data is usually stored as encrypted form in the cloud. At the same time, it also brings a new question that how users can get object encrypted data without decrypting

of them. Searchable encryption is a primitive, which enables data users to search over the encrypted data. Both keywords privacy and data privacy are protected in this procedure. PEKS provides a solution for the third party user to search on remote encrypted data and cloud server can return ciphertext corresponding the user's keywords. Thus, PEKS is suitable for three party situation application in cloud environment.

Considering a scenario: Patients upload the encrypted Personal Health Record (PHR) to the cloud server. Chief physician can search the patient's PHR information by keywords and its private key. PEKS can solve this background problem. But there are still two practical problems not solved by this method. Firstly, since the cloud server may be dishonest, which perform search operation on encrypted data and only return fraction information about the result. Secondly, if the cloud server perform search operation honestly, however, the Chief physician does not recognize the server to perform the search operation correctly on encrypted data. Traditional PEKS schemes can not solve these two questions. So, the PEKS scheme should support verifiable property, allowing the Chief physician can verify the cloud server whether executed the search operation correctly. At the same time, it also has the second question that how cloud server can prove that it performs the search operation honestly when the Chief physician maliciously denies the cloud server's search result. How to design a PEKS scheme to guarantee the confidentiality of PHR data and allow the search results can achieve public verifiability is a challenging problem.

Our contributions: We propose a PEKS with public verifiability scheme in which a data owner can encrypt message and keywords by the user public key, such that only the receiver who has private key can search the keyword in cloud environment and anyone can verify the cloud server whether returned the right result, which not just the data receiver can verify the search result. The most important thing is that the search process and verification process does not leak any information about the query and encrypted data. To the best of our knowledge, although a large body of PEKS with verifiability schemes have been proposed, few works have been done on PEKS scheme with public verifiability. Our scheme can achieve public verifiability while keeping the security of keywords indistinguishability, trapdoor indistinguishability and keyword guessing attack.

## 2 Related Works

Song et al. [1] proposed the first searchable encryption in 2000. This scheme is that the data owner uploads the encrypted data to the cloud server and searches by himself. But it can only support single keyword search and search requires linearly scan each file document word by word. The most important thing is that it is not fully secure and only supports user-server-user model. After this paper, many searchable encryption schemes [2-4] focusing on this model based on symmetric encryption. But these schemes are still unsuitable for three party situation. Symmetric searchable encryption schemes only supports user-server-user model, which is unsuitable in the cloud environment. Boneh et al. [5] proposed the first public key encryption with keyword search (PEKS) in 2004. Their

scheme provides a solution for the third party user to search on remote data encrypted by data owner. However, Boneh's scheme requires a secure channel and can not achieve indistinguishable of trapdoor. Following Boneh's work, Baek et al. [6] proposed the notion of PEKS scheme without secure channel. Park et al. [7] proposed a new security model which is named public key encryption with conjunctive field keyword search. Abdalla et al. [8] proposed a general transformation from identity based encryption (IBE) to PEKS and the definition consistency of searchable encryption.

In order to resist the cloud server's dishonest behavior and return the incorrect search result, Chai and Gong [9] first proposed the concept of verifiable symmetric searchable encryption (VSSE) and given a formal VSSE definition of the protocol, including data owner and cloud server two participants. The data owner uploads the encrypted data, the cloud server performs the search operation, and the data owner receives the data search result and the search result proof. But it can only support single keyword search. Therefore, for improving the function of VSSE, Wang et al. [10] proposed a new VSSE scheme to support fuzzy keyword search. Zheng et al. [11] combines attribute encryption, digital signature, bloom filter, attribute keyword search and proposed a verifiable attribute based keyword search (VABKS) protocol which has good performance in search efficiency, but there are huge computational overhead in the verification process and can not resist offline attack. In the same year, Liu et al. [12] proposed a scheme which compared to the previous scheme, the verification algorithm has been greatly improved in efficiency. However, the Liu's scheme lacks integrity of the keyword detection in the verification process. Generally, the verifier is data owner and the practicality of the VSSE is limited in cloud environment. Many VSSE schemes [13, 14] focusing on this model based on symmetric encryption. All of the above schemes are VSSE schemes and can not support public verifiability.

Alderman et al. [15] made a extension to Parno's verifiable computation scheme [16] from searchable encryption. They proposed a extended functionality in verifiable searchable encryption based on ciphertext policy attribute based encryption. The scheme not only supports more fine-grained keyword expression, but also achieves the public verifiability of search results. Compared to the previous scheme, Alderman's scheme has greatly improved the security and functionality, but the efficiency of the scheme is relatively low. Since the verifier needs to perform verify operation for each file to determine whether to meet the condition of the search query. Moreover, data owner and data receiver need interact with each other, only a data receiver who has private keys from data owner can search and decrypt the ciphertext in cloud environment. Zhang et al. [17] proposed a new public verifiable searchable encryption scheme, although the scheme can not achieve fine-grained search query, but the verification process has been greatly improved. Since the scheme requires secure channel and can not resist offline guessing attack and trapdoor indistinguishable, so the security model and the frame structure of the publicly verifiable searchable encryption are not complete. Meanwhile, Zhang's paper achieves public verifiability by signature for each item of index and files containing keyword, which also brings lower efficiency.

### 3 Preliminaries

In this section, we introduce the formal definition of the bilinear map and complexity assumptions. Let  $G_1$ , and  $G_T$  are two cyclic multiplicative groups of prime order  $p$ . A bilinear map  $e : G_1 \times G_1 \rightarrow G_T$ . Which satisfies:

1. Bilinear: For any  $x, y \in Z_p$ ,  $g \in G_1$ ,  $e(g^x, g^y) = e(g, g)^{xy}$ .
2. Non-degenerate: exist  $g_1, g_2 \in G_1$ ,  $e(g_1, g_2) \neq 1$ .
3. Efficiency: There exists an efficient algorithm to compute  $e(g_1, g_2)$  for all  $g_1, g_2 \in G_1$ .

**Assumption 1** (HDH). The Hash Diffie-Hellman assumption: given the four tuple  $(g, g^x, g^y, H(g^z))$  and hash function  $H$ ,  $x, y, z \in Z_p$ ,  $G_1 = \langle g \rangle$ . It seems that  $x, y$ , and  $z$  is given to the adversary. If the adversary have the  $x, y$ , and  $z$ , decide whether  $z = xy \pmod{p}$  is not hard.

**Assumption 2** (DBDH). The Decisional Bilinear Diffie-Hellman assumption: given the five tuple  $(g, g^x, g^y, g^z, Z)$ ,  $x, y, z \in Z_p$ ,  $Z \in G_T$ ,  $G_1 = \langle g \rangle$ . It seems that  $x, y$ , and  $z$  is given to the adversary. If the adversary have the  $x, y$ , and  $z$ , decide whether  $Z = e(g, g)^{xyz}$  is not hard.

**Assumption 3** (BDH). The Bilinear Diffie-Hellman assumption: given the four tuple  $(g, g^x, g^y, g^z)$ ,  $x, y, z \in Z_p$ ,  $G_1 = \langle g \rangle$ . It seems that  $x, y$ , and  $z$  is given to the adversary. If the adversary have the  $x, y$ , and  $z$ , decide whether compute the value  $(g, g)^{xyz} \in G_T$  is not hard.

**Assumption 4** (BDHI). The Bilinear Diffie-Hellman Inversion assumption: given the two tuple  $(g, g^x)$ ,  $x \in Z_p$ ,  $G_1 = \langle g \rangle$ . It seems that  $(g, g^x)$  is given to the adversary. If the adversary have the  $(g, g^x)$ , decide whether compute the value  $e(g, g)^{\frac{1}{x}} \in G_T$  is not hard.

**Assumption 5** (KEA1-r). The Knowledge of Exponent Assumption: given the three tuple  $(N, g, g^s)$  and returning  $(M, N)$  with  $N = M^s$  to adversary  $\mathcal{A}$ , there exists extractor  $A'$ , which given the same input as  $\mathcal{A}$  returns  $d$  such that  $M = g^d$ .

**Assumption 6** (DL). The Discrete logarithm assumption: given the two tuple  $(g, R)$ ,  $x \in Z_p$ ,  $G_1 = \langle g \rangle$ ,  $R \in G_1$ . It seems that  $(g, R)$  is given to the adversary. If the adversary have the  $(g, R)$ , decide whether find the only integer  $x$  which satisfy  $g^x = R \pmod{p}$  is not hard.

## 4 PEKS

### 4.1 PEKS Model

Now, we will discuss the PEKS model. We consider the public-key scenario in which there are a data owner, an untrusted server, a data receiver, anyone verifier. The data owner encrypts index and data with receiver's public key and cloud server public key, the receiver can send the trapdoor to the server with

the receiver's secret key, so that the cloud server can perform search operation on encrypted data and data owner successfully authorizes data receiver's search ability through a public channel. A general PEKS scheme include five algorithms: Setup algorithm, Keygen algorithm, PEKS algorithm, Trapdoor algorithm, Test algorithm.

- o **Setup**( $1^k$ )  $\rightarrow gp$ : This algorithm inputs a security parameter  $1^k$ , and generates a global parameter  $gp$ .
- o **KeyGen**( $gp$ )  $\rightarrow (pk_s, sk_s, pk_r, sk_r)$ : This algorithm inputs a global parameter  $gp$ , and generates two pairs of public and secret keys  $(pk_r, sk_r), (pk_s, sk_s)$  for receiver and server respectively.
- o **PEKS**( $gp, pk_r, pk_s, w$ )  $\rightarrow C_w$ : The data owner inputs global parameter  $gp$ , the receiver public key  $pk_r$ , the server public key  $pk_s$ , and the keyword  $w$ , outputs a ciphertext  $C_w$  for  $w$ .
- o **Trapdoor**( $gp, pk_s, sk_r, w$ )  $\rightarrow T_w$ : The data receiver inputs keyword  $w$ , the server public key  $pk_s$ , the receiver secret key  $sk_r$ , the global parameter  $gp$ , and computes trapdoor  $T_w$  corresponding the keyword  $w$ .
- o **Test**( $T_w, C_w, sk_s$ )  $\rightarrow C_w$  or  $\perp$ : The server inputs the ciphertext  $C_w$ , a trapdoor  $T_w$  and the server secret key  $sk_s$ . It outputs the ciphertext  $C_w$  if  $w = w'$ , and  $\perp$  otherwise.

## 4.2 Security Model

**Definition 1** (Chosen keyword attack). A PEKS scheme satisfies ciphertext indistinguishability secure against chosen keyword attack if for any probability polynomial time adversary  $\mathcal{A}$ , there is a neglible function  $\epsilon(\lambda)$  such that

$$Adv_{\mathcal{A}, \{i \in 1, 2\}}^{cka} = |Pr[b = b'] - 1/2| \leq \epsilon(\lambda).$$

A PEEKS scheme, we can define by the ciphertext indistinguishability experiment as follows:

*Game1*

- o **Setup**:  $\mathcal{A}_1$  is assumed to be a malicious server. The public parameter  $gp$ , the server key pairs  $(pk_s, sk_s)$  and the receiver public key  $pk_r$  are given to the  $\mathcal{A}_1$ .
- o **Phase 1-1**:  $\mathcal{A}_1$  makes the queries of the trapdoor  $T_w$ , adaptively makes any keyword queries for  $w \in \{0, 1\}^*$ .
- o **Phase 1-2**:  $\mathcal{A}_1$  gives challenger  $\mathcal{B}$  two be challenged keywords,  $w_0$  and  $w_1$ .  $\mathcal{B}$  randomly picks bit  $b$ , and computes a ciphertext for  $w_b$  and returns it to  $\mathcal{A}_1$ .
- o **Phase 1-3**:  $\mathcal{A}_1$  continues making trapdoor queries of the form  $w$  and the attacker can not ask for the trapdoors  $w_0$  and  $w_1$ .
- o **Phase 1-4**:  $\mathcal{A}_1$  outputs its guess  $b'$ .

In this attack experiment, the advantage of the adversary  $\mathcal{A}_1$  is:

$$Adv_{\mathcal{A}_1}^{cka} = |Pr[b = b'] - 1/2|.$$

*Game2*

- **Setup:**  $\mathcal{A}_2$  is assumed to be a outside attacker. The public parameter  $gp$ , the receiver key pair  $(pk_r, sk_r)$  and the server public key  $pk_s$  are given to the  $\mathcal{A}_2$ .
- **Phase 1-1:**  $\mathcal{A}_2$  makes the queries of the trapdoor  $T_w$ , adaptively makes any keyword queries for  $w \in \{0, 1\}^*$ .
- **Phase 1-2:**  $\mathcal{A}_2$  gives challenger  $\mathcal{B}$  two be challenged keywords,  $w_0$  and  $w_1$ .  $\mathcal{B}$  randomly picks bit  $b$ , and computes a ciphertext for  $w_b$  and returns it to  $\mathcal{A}_2$ .
- **Phase 1-3:**  $\mathcal{A}_2$  continues making trapdoor queries of the form  $w$  and the attacker can not ask for trapdoors  $w_0$  and  $w_1$ .
- **Phase 1-4:**  $\mathcal{A}_2$  outputs its guess  $b'$ .

About PEKS scheme, we can define the trapdoor indistinguishability experiment as follows:

*Game3*

- **Setup:**  $\mathcal{A}$  is assumed to be an polynomial time attack algorithm, running time is bounded by  $t$ , the global parameter  $gp$ , the server public key  $pk_s$  and the receiver public key  $pk_r$  are given to the  $\mathcal{A}$ , keeping  $sk_r, sk_s$  secret.
- **Phase 1-1:**  $\mathcal{A}$  makes the queries of the trapdoor  $T_w$ , adaptively makes any keyword queries for  $w \in \{0, 1\}^*$ .
- **Phase 1-2:**  $\mathcal{A}$  gives challenger  $\mathcal{B}$  two be challenged keywords  $w_0$  and  $w_1$ .  $\mathcal{B}$  randomly picks bit  $b$ , and computes a trapdoor  $T_{w_b}$  for  $w_b$  and returns it to  $\mathcal{A}$ .
- **Phase 1-3:**  $\mathcal{A}$  continues making trapdoor queries of the form  $w$  and the attacker can not ask for trapdoors  $w_0$  and  $w_1$ .
- **Phase 1-4:**  $\mathcal{A}$  outputs its guess  $b'$ .

The advantage of the adversary  $\mathcal{A}$  in this game is defined as:

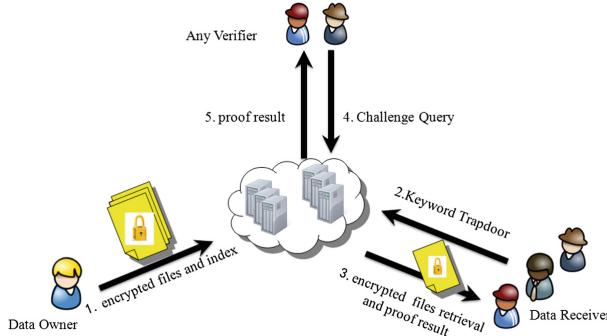
$$Adv_{\mathcal{A}}^{Trapdoorindistinguishability} = |Pr[b = b'] - 1/2|.$$

**Definition 2** (Trapdoor indistinguishability). A PEKS scheme is trapdoor indistinguishability secure for any probability polynomial time adversary  $\mathcal{A}$ , there is a negligeable function  $\epsilon(\lambda)$  such that

$$Adv_{\mathcal{A}}^{Trapdoorindistinguishability} = |Pr[b = b'] - 1/2| \leq \epsilon(\lambda).$$

## 5 A PEKS with Public Verifiability Scheme

In this section, we will propose a general construction method and security reduction for PEKS with public verifiability through study of PEKS. Data owners encrypt the keyword set with the receiver public key  $pk_r$  and cloud server public key  $pk_s$ . Data receiver generates the trapdoor with the receiver secret key  $sk_r$  and cloud server public key  $pk_s$ , and server matches the ciphertext with trapdoor and returns partial ciphertext file corresponding the keyword, any verifier can check the search result from server. Figure 1 shows the entire system framework. A PEKS with public verifiability scheme consists algorithms as follows:



**Fig. 1.** PEKS with public verifiability

- o **Setup**( $1^k$ )  $\rightarrow gp$ : This algorithm inputs a security parameter  $1^k$ , and generates a global parameter  $gp$ .
- o **KeyGen**( $gp$ )  $\rightarrow (pk_s, sk_s, pk_r, sk_r)$ : This algorithm inputs a global parameter  $gp$ , and generates two pairs of public and secret keys  $(pk_r, sk_r)$ ,  $(pk_s, sk_s)$  for receiver and server respectively.
- o **TagGen**( $gp, pk_r, w$ )  $\rightarrow F_w$ : The data owner inputs the receiver public key  $pk_r$ , the keyword  $w$  and computes the keyword tag  $F_w$ , releases  $F_w$  to be publicly known to everyone.
- o **PEKS**( $gp, pk_r, pk_s, w$ )  $\rightarrow C_w$ : The data owner inputs global parameter  $gp$ , the receiver public key  $pk_r$ , the server public key  $pk_s$ , and the keyword  $w$ , outputs a ciphertext  $C_w$  for  $w$ .
- o **Trapdoor**( $gp, pk_s, sk_r, w$ )  $\rightarrow (T_w, ch_w)$ : The data receiver inputs keyword  $w$ , the server public key  $pk_s$ , the receiver secret key  $sk_r$ , the global parameter  $gp$ , and computes trapdoor  $T_w$  and the challenge query  $ch_w$  corresponding the keyword  $w$ .
- o **Test**( $T_w, C_w, sk_s, ch_w$ )  $\rightarrow C_w$  or  $\perp$ : The server inputs the ciphertext  $C_w$ , a trapdoor  $T_w$ , the challenge query  $ch_w$  and the server secret key  $sk_s$ . It outputs the ciphertext  $C_w$  and the challenge response  $R$ , if  $w = w'$ ; and  $\perp$  otherwise.
- o **Private-Verify**( $T_w, R, gp$ )  $\rightarrow 1$  or  $0$ : The data receiver inputs the global parameter  $gp$ , the challenge response  $R$ , and the return ciphertext  $C_w$ . It outputs the result 1, if the server performs search operation honestly, and 0 otherwise.
- o **Public-Verify**( $T_w, F_w, gp$ )  $\rightarrow 1$  or  $0$ : For public verifiability, we can divide into three steps:

**Challenge query**( $gp$ )  $\rightarrow ch_w$ : In order to verify the correctness file corresponding the keyword  $w$ , the public verifier inputs the global parameter  $gp$ . It outputs the challenge query  $ch_w$ .

**GenProof**( $ch_w, gp, C_w$ )  $\rightarrow R$ : The server inputs the return ciphertext  $C_w$ , the global parameter  $gp$ , the challenge query  $ch_w$ . It outputs the challenge response  $R$ .

**Verify**( $F_w, R$ )  $\rightarrow 1$  or  $0$ : The public verifier inputs the keyword tag  $F_w$ , the

challenge response  $R$ . It outputs the result 1, if the server performs search operation honestly, and 0 otherwise.

About PEKS with public verifiability scheme, we can define the public verifiability security experiment as follows:

*Game4*

- **Setup:**  $\mathcal{A}$  is assumed to be an polynomial time attack algorithm and a malicious server, running time is bounded by  $t$ , the global parameter  $gp$ , the server key pair  $(pk_s, sk_s)$  and the receiver public key  $pk_r$  are given to the  $\mathcal{A}$ , keeping  $sk_r$  secret.
- **Phase 1-1:**  $\mathcal{A}$  makes the queries of the trapdoor  $T_w$  and the challenge query  $ch_w$ , adaptively makes any keyword  $w$ . The challenger computes a keyword tag  $F_w$  and sends to the adversary.
- **Phase 1-2:**  $\mathcal{A}$  outputs the ciphertext  $C'_w$  and the forge challenge response  $R'$ .
- **Phase 1-3:** The challenger outputs a bit  $b = 1$ , if the verification process can pass successfully, and 0 otherwise.

The advantage of the adversary  $\mathcal{A}$  in this game is defined as:

$$Adv_{\mathcal{A}}^{Public\ Verifiability} = |Pr[b = 1] - 1/2|.$$

**Definition 3** (Public verifiability). A PEKS with public verifiability scheme is public verifiability secure for any probability polynomial time adversary  $\mathcal{A}$ , there is a neglible function  $\epsilon(\lambda)$  such that

$$Adv_{\mathcal{A}}^{Public\ Verifiability} = |Pr[b = 1] - 1/2| \leq \epsilon(\lambda).$$

Since the verifier is not the data receiver himself, the scheme should ensure that any verifier can not get the private information about keywords and data file. We can define the security against any verifier by the simulation paradigm. Let  $f$  be an probabilistic polynomial time functionality and let  $\Pi$  be a two-party protocol for computing  $f$ . We denote the verifier  $V$ .

**Definition 4** (Privacy against semi honest behavior [18]).  $f$  is a deterministic functionality,  $\Pi$  be a two-party protocol for computing  $f$  privately, if there exist probabilistic polynomial time algorithms, denoted  $S_1$  and  $S_2$ .

$$\{S_1(x, f_1(x, y))\}_{x,y \in \{0,1\}^*} \stackrel{c}{=} \{View_1^\Pi(x, f_1(x, y))\}_{x,y \in \{0,1\}^*}.$$

$$\{S_2(x, f_V(x, y))\}_{x,y \in \{0,1\}^*} \stackrel{c}{=} \{View_2^\Pi(x, f_2(x, y))\}_{x,y \in \{0,1\}^*}.$$

By Definition 4, we can define the privacy against any verifier similarity, which is given in Definition 5.

**Definition 5** (Privacy against any verifier). A PEKS with public verifiability scheme is privacy against any verifier, for the keyword integrity checking protocol  $\Pi$ , if there exists a probability polynomial time simulator  $S_v$  such that  $\{S_v(x, f_V(x, y))\}_{x,y \in \{0,1\}^*} \stackrel{c}{=} \{View_V^\Pi(x, f_V(x, y))\}_{x,y \in \{0,1\}^*}$ .

## 6 A Concrete PEKS with Public Verifiability Scheme

### 6.1 Scheme Description

We will give a concrete PEKS with public verifiability scheme that is based on PEKS scheme. This scheme consists algorithms as follows:

- o **Setup**( $1^k$ )  $\rightarrow gp$ : This algorithm inputs a security parameter  $1^k$ , and generates a global parameter  $gp = (N, g, g_1, \eta, \sigma, H, H_1, H_2, G_1, G_T, f)$ , letting  $N = p_1 q_1$ ,  $p_1 = 2p' + 1$ ,  $q_1 = 2q' + 1$  are two large primes,  $p'$  and  $q'$  are primes,  $QR_N$  denotes the quadratic residues multiplicative cyclic group, which the generator is  $g_1$ , the order of  $g_1$  is  $p'q'$ .  $G_1$  and  $G_T$  are two cyclic multiplicative groups of prime order  $p$ ,  $g \in G_1$ , several random elements  $\eta, \sigma \in G_1$ ,  $H : \{0, 1\}^* \rightarrow G_1$ ,  $H_1 : \{0, 1\}^* \rightarrow G_1$ ,  $H_2 : G_T \rightarrow \{0, 1\}^k$ ,  $f$  is a pseudo-random function,  $f : \{0, 1\}^k \times \{0, 1\}^{\log_2^l} \rightarrow \{0, 1\}^d$ ,  $d$  is a security parameter.
- o **KeyGen**( $gp$ )  $\rightarrow (pk_s, sk_s, pk_r, sk_r)$ : This algorithm inputs a global parameter  $gp$ , chooses random numbers  $\alpha$  and generates two pairs of public and secret key  $(pk_r, sk_r)$ ,  $(pk_s, sk_s)$  for receiver and server respectively, which  $pk_r = (pk_{r_1}, pk_{r_2}) = (g^\beta, \eta^\beta)$ ,  $sk_r = (\beta, p_1, q_1)$ ,  $pk_s = (pk_{s_1}, pk_{s_2}) = (g^\alpha, \sigma^{\frac{1}{\alpha}})$ ,  $sk_s = \alpha$ .
- o **TagGen**( $gp, j, w_i$ )  $\rightarrow (F_w)$ : The data owner inputs global parameter  $gp$ , the keyword  $w_i, i \in \{1, 2 \dots n\}$ , the file index number is  $j$  corresponding the keyword  $w_i$  and computes the keyword tag  $F_{w_i} = \{F_{w_{ij}} = g_1^{H(w_{ij})} \text{mod } N\} = \{g_1^{H(w_i||j)} \text{mod } N\}, j \in \{1, 2 \dots l\}$ , releases  $F_w$  to be publicly known to everyone.
- o **PEKS**( $gp, pk_r, pk_s, w$ )  $\rightarrow C_w$ : The data owner inputs global parameter  $gp$ , the receiver public key  $pk_r$ , the server public key  $pk_s$ , the keyword  $w_i$ , and chooses random numbers  $r \in Z_{p^*}$  outputs a ciphertext  $C_{w_i} = [A, B, C] = [pk_{r_1}^r, H_2(e(H_1(w_i)^r, pk_{s_1})), F_{w_{ij}}]$  for  $w_i$ .
- o **Trapdoor**( $gp, pk_s, sk_r, w$ )  $\rightarrow (T_w, ch_w)$ : The data receiver inputs keyword  $w_i$ , the server public key  $pk_s$ , the receiver secret key  $sk_r$ , the global parameter  $gp$ , chooses random numbers  $r' \in Z_p, t \in [1, 2^k - 1], s \in Z_N \setminus \{0\}$  and computes trapdoor  $T_w = [T_1, T_2] = [g^{r'}, H_1(w)^{\frac{1}{\beta}} \cdot H(pk_{s_1}^{r'})]$  and the challenge query  $ch_w = < t, g_s = g_1^s \text{mod } N >$  corresponding the keyword  $w$ .
- o **Test**( $T_w, C_w, sk_s, ch_w$ )  $\rightarrow C_w$  or  $\perp$ : The server inputs the ciphertext  $C_w$ , a trapdoor  $T_w$  and the server secret key  $sk_s$ . If  $B = H_2(e(A, (\frac{T_2}{H(T_1^{sk_s})})^{sk_s}))$ , it outputs the ciphertext  $C_{w_{ij}}$  and computes the coefficient  $a_j = f_t(j)$ , outputs the challenge response  $R = (g_s)^{\sum_{j=1}^l a_j F_{w_{ij}}} \text{mod } N$ , and  $\perp$  otherwise.
- o **Private-Verify**( $T_w, gp, R$ )  $\rightarrow 1$  or  $0$ : The data receiver inputs the global parameter  $gp$ , the challenge response  $R$  and the return ciphertext  $C_w$ . It outputs the result 1, if the server honestly perform search operation which has  $R' = g_1^{(\sum_{j=1}^l a_j F_{w_{ij}})^s} \text{mod } N$  and  $R = R'$ , and 0 otherwise.
- o **Public-Verify**( $T_w, pk_r, F_w, gp$ )  $\rightarrow 1$  or  $0$ : For public verifiability, we can divide into three steps:

**Public-challenge query**( $gp$ )  $\rightarrow ch_w$ : In order to verify the correctness file corresponding the keyword  $w$ , the public verifier inputs the global parameter  $gp$ , chooses random number  $t \in [1, 2^k - 1], s \in Z_N \setminus \{0\}$ . It outputs the challenge query  $ch_w = < t, g_s = g_1^s \text{ mod } N >$ .

**GenProof**( $ch_w, gp, C_w$ )  $\rightarrow R$ : The server inputs the return ciphertext  $C_w$ , the global parameter  $gp$ , the challenge query  $ch_w$ . It outputs the challenge response  $R = (g_s)^{\sum_{j=1}^l a_j F_{w_{ij}}} \text{ mod } N$ .

**Verify**( $F_w, R, gp$ )  $\rightarrow 1$  or  $0$ : The public verifier inputs the global parameter  $gp$ , the keyword tag  $F_{w_{ij}}$ , the challenge response  $R$ . It outputs the result  $1$ , if the server honestly perform search operation which has  $R' = g_1^{(\sum_{j=1}^l a_j F_{w_{ij}})s} \text{ mod } N$  and  $R = R'$ , and  $0$  otherwise.

**Correctness:** When assuming the ciphertext is valid for  $W'$  and the trapdoor  $T_W$  for  $W$ , we can verify query correctness as:  $H_2(e(H_1(w_i)^r, pk_{s_1})) = H_2(e(A, (\frac{T_2}{H(T_1^{sk_s})})^{sk_s})$  test whether two values  $W$  and  $W'$  are equal. For the public verification correctness, we can verify correctness as:

$R' = g_1^{(\sum_{j=1}^l a_j F_{w_{ij}})s} \text{ mod } N = R$ , test whether two values  $R$  and  $R'$  are equal. Since the verifier can compute the coefficient  $a_j = f_t(j)$  and the  $F_w$  to be publicly known. So, the verifier can verify the protocol correctness whether the server honestly perform protocol, include the server returns an empty set.

## 7 Security and Performance

### 7.1 Security Proof

In this section, we will prove the security of the protocol against the untrusted server and malicious receiver similar the paper [21–23]. Theorems guarantee that if the server return the complete search results, it can pass the private and public verifiability successfully.

**Theorem 1:** Suppose the BDH and BDHI problem is hard, the PEKS with public verifiability scheme is secure under selective keyword model attack.

*Proof.* About selective keyword attack, we need to resist two adversaries which are a malicious server adversary  $\mathcal{A}_1$  and malicious outside adversary  $\mathcal{A}_2$ . We will proof the theorem similar paper [19], we will give the proof in the extended version, please see the full version of this paper.

**Theorem 2:** PEKS with public verifiability scheme is an secure scheme satisfies the trapdoor indistinguishability against a chosen keyword attack, under assumption that Hash Diffie-Hellman (HDH) is intractable.

*Proof.* Security proof similar paper [19], we will give the proof in the extended version, please see the full version of this paper.

**Theorem 3:** PEKS with public verifiability scheme is an secure scheme against the untrusted server, under the KEA1-r and the large integer factorization assumption.

*Proof.* Adversary  $\mathcal{A}$  is an polynomial time attack algorithm can break the PEKS with public verifiability scheme with the advantage  $\varepsilon$ . We construct an algorithm  $\mathcal{B}$  that solves the integer factorization problem with  $\varepsilon'$ . Algorithm  $\mathcal{B}$  is given a larger integer  $N = p_1 q_1$ . It's goal is to output two large prime number  $p_1$  and  $q_1$ . Algorithm  $\mathcal{B}$  simulates the challenger and interacts with adversary  $\mathcal{A}$ . Since the space restrictions, the simulation will in the full version.

Adversary  $\mathcal{A}$  forge an requested keyword response  $R$  to the algorithm  $\mathcal{B}$ . If the verify  $(F_w, R, pk_r, gp)$  can pass successfully, the adversary  $\mathcal{A}$  forge  $R$  successfully. Let we analyze the integer factorization problem. Adversary  $\mathcal{A}$  has the tuple  $(N, g, g^s)$  and outputs the response  $R = (g_s)^{\sum_{j=1}^l a_j F_{w_{ij}}}, a_j = f_t(j)$  for  $j \in \{1, 2 \dots l\}$ . Because  $\mathcal{A}$  can naturally computes  $P = g_1^{\sum_{j=1}^l a_j F_{w_{ij}}}$ , so  $\mathcal{A}$  has two tuple  $(R = P^s, P)$ . Since the knowledge of exponent assumption, we can construct an extractor  $\mathcal{A}'$  and output  $d$  which satisfy  $P = g^d \bmod N$ , Algorithm  $\mathcal{B}$  can obtain  $d = \sum_{n=1}^j a_j H(w_i \parallel j) \bmod p'q'$ .

Since the space restrictions, the equation solution will in the full version.

By solving the above equation, algorithm  $\mathcal{B}$  can get  $H^*(w_{ij}) = H(w_{ij}) \bmod p'q'$ , for each  $j \in \{1, 2 \dots l\}$ . If  $H^*(w_{ij}) = H(w_{ij})$ , algorithm  $\mathcal{B}$  can extract all the keyword Hash function  $H(w_{ij})$ , otherwise, there exist  $j$ ,  $H^*(w_{ij}) \neq H(w_{ij})$ , then algorithm  $\mathcal{B}$  can compute the integer prime factorization about  $N$ . Because  $H^*(w_{ij}) = H(w_{ij}) \bmod p'q'$ , algorithm  $\mathcal{B}$  can obtain a multiple of  $\phi(N)$ , so  $\mathcal{B}$  can compute the integer prime factorization about  $N$  from the lemma 1 in [20]. Because of the difficulty of the integer prime factorization, we can see that any keyword Hash function can be extracted. Overal, the proposed scheme guarantees the keyword integrity against an untrusted server.

**Theorem 4:** PEKS with public verifiability scheme is an secure scheme against the third party verifier, under the semi-honest model.

*Proof.* In this proof, we can prove the scheme security against the third party verifier by a two party protocol for computing in the semi-honest model. Security proof similar paper [21]. The verifier and the server are denoted by  $V$  and  $P$  respectively. The view of the third party verifier is denoted as  $View_V^P$ . Exists a probability polynomial time simulator  $S_v$  for the view of the verifier. Our goal is to prove that the output of the simulator  $S_v$  is computationally indistinguishable the  $View_V^P$  of the verifier. The verifier has the tuple  $(N, g, \{F_{w_i, i=1,2,\dots,n}\})$  as the input and output a bit  $b$  as result which denotes the success or failure. The simulator's input and output is similar with the verifier. If the server is honest, the bit  $b$  is always 1. Since the space restrictions, the simulation will in the full version.

In conclusion, we have the formula  $\{S_v(x, f_V(x, y))\} \stackrel{c}{=} \{View_V^P(x, f_V(x, y))\}$  is proved,  $x, y \in \{0, 1\}^*$ . So the verifier cannot get any information from the search result, except its input and output.

**Theorem 5:** PEKS with public verifiability scheme is an secure scheme which can achieve public verifiability, under assumption that DL is intractable.

*Proof.* To proof our PEKS with public verifiability scheme is an secure scheme which can achieve public verifiability, we can discuss that the adversary  $\mathcal{A}$  forge an requested keyword response  $R$  and verify  $(F_w, R, gp) = 1$ . According to equation  $R = R'$ , we can compute the equation probability  $\Pr[(g_s)^{\sum_{j=1}^l a_j F_{w_{ij}}} = g_1^{(\prod_{j=1}^l (a_j F'_{w_{ij}}))s}]$ . Let's simplify it further,  $\Pr[H(w'_i \parallel j) = H(w_i \parallel j)], j = 1, 2 \dots l$ . Since the  $H$  is a collision-resistant Hash function, according to the Hash function definition, the adversary can not find a keyword  $w'_i$  such that  $w'_i \neq w_i$  and  $H(w'_i \parallel j) = H(w_i \parallel j), j = 1, 2 \dots l$ . Besides the equation  $\Pr[H(w'_i \parallel j) = H(w_i \parallel j)], j = 1, 2 \dots l$ , the adversary need extractor the  $H(w'_i \parallel j)$  from  $g^{H(w'_i \parallel j)}$  is also a hard problem in DL assumption.

About the security keyword guessing attack, we add public key and private key for the cloud server and the data owner can re-encrypt the keyword ciphertext and trapdoor in public channel. This guarantees only the cloud server can match the keyword ciphertext and trapdoor. Our scheme can effectively prevent the guessing attack by this approach.

## 7.2 Security and Performance Analysis

Basic operations are recorded as: Let  $E$  denote an exponentiation operation,  $P$  is the basic operation of hash operation,  $M$  denote a multiplication operation in the group,  $e$  denote a pairing operation,  $f$  denote a polynomial operation and  $k$  represent the maximum number of trapdoor. Tables 1 and 2 give us the comparison between our scheme and the previous public key encryption with keyword search scheme. We use Trap Ind, Ciph Ind, PV, KS, KR to denote Trapdoor indistinguishability, Ciphertext indistinguishability, Public verifiability, Keyword guessing attack, KeyGenServer, KeyGenReceiver.

**Table 1.** Security comparison

	Boneh et al.	Baek et al.	Yang et al.	Our.
Trap Ind	No	No	Yes	Yes
Ciph Ind	Yes	Yes	Yes	Yes
KG	No	No	Yes	Yes
PV	No	No	No	Yes

**Table 2.** Performance comparison

	Boneh et al.	Baek et al.	Yang et al.	Our.
KS	-	M	-	2E
KR	E	M	6E	2E
PEKS	$2E + 2P + e$	$E + M + P + 2e$	$(2k + 6)E$	$2E + P + e$
Trapdoor	$E + P$	$P + M$	$4f$	$4E + 2P + M$
Test	$e + P$	$M + e$	$2E + 4f$	$1f + 3E + P + e$

## 8 Conclusions

In order to overcome the disadvantages of traditional PEKS in public verifiability environment, this paper proposes a PEKS with public verifiability scheme by using bilinear technique. By the security analysis, we have proved that the scheme achieves keyword indistinguishability, trapdoor indistinguishability, keyword guessing attack, public verifiability. Comparing with existing schemes, our scheme supports the public verifiability in cloud environment. It is a suitable method for solving the practical problem which is described in the introduction. Through the aforementioned content, we can get that this proposed the public verifiability of public key encryption with keyword search scheme is a secure and wide applicable protocol, and has a certain practical value.

**Acknowledgment.** This work is supported by the National Nature Science Foundation of China under Grant No: 61272091 and No: 61772311.

## References

1. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: 2000 IEEE Symposium on Security and Privacy, pp. 44–55. IEEE Computer Society (2000)
2. Goh, E.J.: Secure indexes. IACR Cryptol. ePrint Arch. **2003**, 216 (2003)
3. Reza, C., Garay, J., Kamara, S., Ostrovsky, R.: Searchable symmetric encryption: improved definition and efficient constructions. In: 2006 Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 79–88. ACM (2006)
4. Chang, Y.-C., Mitzenmacher, M.: Privacy preserving keyword searches on remote encrypted data. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 442–455. Springer, Heidelberg (2005). [https://doi.org/10.1007/11496137\\_30](https://doi.org/10.1007/11496137_30)
5. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-24676-3\\_30](https://doi.org/10.1007/978-3-540-24676-3_30)
6. Baek, J., Safavi-Naini, R., Susilo, W.: Public key encryption with keyword search revisited. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008. LNCS, vol. 5072, pp. 1249–1259. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-69839-5\\_96](https://doi.org/10.1007/978-3-540-69839-5_96)
7. Park, D.J., Kim, K., Lee, P.J.: Public key encryption with conjunctive field keyword search. In: Lim, C.H., Yung, M. (eds.) WISA 2004. LNCS, vol. 3325, pp. 73–86. Springer, Heidelberg (2005). [https://doi.org/10.1007/978-3-540-31815-6\\_7](https://doi.org/10.1007/978-3-540-31815-6_7)
8. Abdalla, M., Bellare, M., Catalano, D., et al.: Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extensions. J. Cryptol. **21**(3), 350–391 (2008)
9. Chai, Q., Gong, G.: Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers. In: 2012 IEEE International Conference on Communications, pp. 917–922. IEEE (2012)

10. Wang, J., Ma, H., Tang, Q., et al.: Efficient verifiable fuzzy keyword search over encrypted data in cloud computing. *Comput. Sci. Inf. Syst.* **10**(2), 667–684 (2013)
11. Zheng, Q., Xu, S., Ateniese, G.: VABKS: verifiable attribute-based keyword search over outsourced encrypted data. In: INFOCOM, 2014 Proceedings IEEE, pp. 522–530. IEEE (2014)
12. Liu, P., Wang, J., Ma, H., et al.: Efficient verifiable public key encryption with keyword search based on KP-ABE. In: 2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), pp. 584–589. IEEE (2014)
13. Wei, X., Zhang, H.: Verifiable multi-keyword fuzzy search over encrypted data in the cloud. In: 2016 International Conference on Advanced Materials and Information Technology Processing, pp. 271–277 (2016)
14. Nie, X., Liu, Q., Liu, X., Peng, T., Lin, Y.: Dynamic verifiable search over encrypted data in untrusted clouds. In: Carretero, J., Garcia-Blas, J., Ko, R.K.L., Mueller, P., Nakano, K. (eds.) ICA3PP 2016. LNCS, vol. 10048, pp. 557–571. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-49583-5\\_44](https://doi.org/10.1007/978-3-319-49583-5_44)
15. Alderman, J., Janson, C., Martin, K.M., Renwick, S.L.: Extended functionality in verifiable searchable encryption. In: Pasalic, E., Knudsen, L.R. (eds.) Balkan-CryptSec 2015. LNCS, vol. 9540, pp. 187–205. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-29172-7\\_12](https://doi.org/10.1007/978-3-319-29172-7_12)
16. Parno, B., Raykova, M., Vaikuntanathan, V.: How to delegate and verify in public: verifiable computation from attribute-based encryption. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 422–439. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-28914-9\\_24](https://doi.org/10.1007/978-3-642-28914-9_24)
17. Zhang, R., Xue, R., Yu, T., et al.: PVSAE: a public verifiable searchable encryption service framework for outsourced encrypted data. In: 2016 IEEE International Conference on Web Services, pp. 428–435. IEEE (2016)
18. Goldreich, O.: Foundations of Cryptography. Cambridge University Press, Cambridge (2004)
19. Rhee, H.S., Park, J.H., Susilo, W., et al.: Trapdoor security in a searchable public-key encryption scheme with a designated tester. *J. Syst. Softw.* **83**(5), 763–771 (2010)
20. Miller, G.L.: Riemann's hypothesis and tests for primality. *J. Comput. Syst. Sci.* **13**(3), 300–317 (1976)
21. Hao, Z., Zhong, S., Yu, N.: A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability. *IEEE Trans. Knowl. Data Eng.* **23**(9), 1432–1437 (2011)
22. Alabdulatif, A., Kumarage, H., Khalil, I., et al.: Privacy-preserving anomaly detection in cloud with a lightweight homomorphic approach. *J. Comput. Syst. Sci.* **90**, 28–45 (2017)
23. Kumarage, H., Khalil, I., Alabdulatif, A., et al.: Secure data analytics for cloud-integrated internet of things applications. *IEEE Cloud Comput.* **3**(2), 46–56 (2016)



# Malicious Bitcoin Transaction Tracing Using Incidence Relation Clustering

Baokun Zheng<sup>1,2</sup>, Liehuang Zhu<sup>1</sup>, Meng Shen<sup>1(✉)ID</sup>, Xiaojiang Du<sup>3</sup>, Jing Yang<sup>1</sup>, Feng Gao<sup>1</sup>, Yandong Li<sup>1</sup>, Chuan Zhang<sup>1</sup>, Sheng Liu<sup>4</sup>, and Shu Yin<sup>4</sup>

<sup>1</sup> Beijing Institute of Technology, Beijing, China  
zhengbk168@163.com, {liehuangz,shenmeng,jingy,leeyandong}@bit.edu.cn,  
gaoｆengbit@foxmail.com, chuanzdlut@163.com

<sup>2</sup> China University of Political Science and Law, Beijing, China  
<sup>3</sup> Temple University, Philadelphia, USA  
dxj@ieee.org

<sup>4</sup> Union Mobile Financial Technology Co., Ltd., Beijing, China  
{liusheng,yinshu}@umfintech.com

**Abstract.** Since the generation of Bitcoin, it has gained attention of all sectors of the society. Law breakers committed crimes by utilizing the anonymous characteristics of Bitcoin. Recently, how to track malicious Bitcoin transactions has been proposed and studied. To address the challenge, existing solutions have limitations in accuracy, comprehensiveness, and efficiency. In this paper, we study Bitcoin blackmail virus WannaCry event incurred in May 2017. The three Bitcoin addresses disclosed in this blackmail event are only restricted to receivers accepting Bitcoin sent by victims, and no further transaction has been found yet. Therefore, we acquire and verify experimental data by example of similar Bitcoin blackmail virus CryptoLocker occurred in 2013. We focus on how to track malicious Bitcoin transactions, and adopt a new heuristic clustering method to acquire incidence relation between addresses of Bitcoin and improved Louvain clustering algorithm to further acquire incidence relation between users. In addition, through a lot of experiments, we compare the performance of our algorithm with another related work. The new heuristic clustering method can improve comprehensiveness and accuracy of the results. The improved Louvain clustering algorithm can increase working efficiency. Specifically, we propose a method acquiring internal relationship between Bitcoin addresses and users, so as to make Bitcoin transaction deanonymisation possible, and realize a better utilization of Bitcoin in the future.

**Keywords:** Bitcoin · Blockchain · Incidence relation · Cluster

## 1 Introduction

On May 12, 2017, Bitcoin blackmail virus WannaCry was burst globally. Criminals blackmailed Bitcoin [1] equaling to USD 300 from users infected with the

virus. For a short while, many users in the world suffered from serious loss. Bitcoin can be sent by anyone to any other person everywhere. Bitcoin uses a public key based wallet address as a pseudonym on the blockchain, where transactions between different users are realized through this pseudonym. Bitcoin accounts are anonymous and cannot be reviewed. In order to implement anonymous transactions, Bitcoin system allows users to generate multiple wallets addresses freely. Users can use different wallet addresses to reduce the transaction characteristics of individual wallet addresses. Because of the anonymity of Bitcoin account, Bitcoin may be used for some illegal behavior and the black market, such as the purchase of guns and drugs. Thus, obtaining trading rules by analyzing the user transaction records, and even speculating the identity information of users, is particularly important in the prevention of crime.

[2–9] studied the relationship between the Bitcoin addresses based on the heuristic method. However, the comprehensiveness of the heuristic methods is not fully considered in these papers. They did not study the output addresses of coinbase transaction, and the judgement on change address is insufficient. In addition, few papers have studied the relation between users.

Under such background, this paper aims to better known traceability of Bitcoin movement and explore better use in the future. Most importantly, this paper does not aim to carry out deanonymisation for all Bitcoin users because it is impossible according to abstract user protocol design. Instead, this paper aims to recognize anonymous users according to specific behaviors of Bitcoin users in Bitcoin network.

In this paper, the methodology is based on the availability of Bitcoin blockchain, using digital signature secret key disclosed on every transaction, and decoding a graphic data structure by Bitcoin activities. To summarize, the ***contributions*** of our work include:

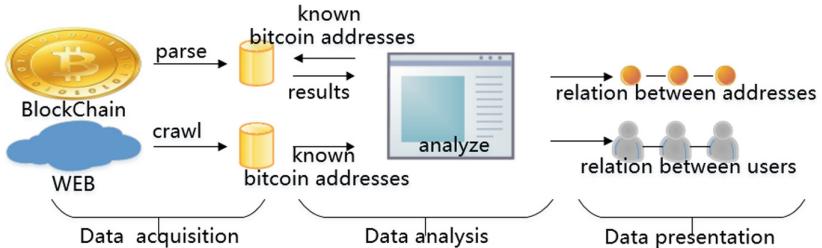
- (1) We propose a new heuristic method with three rules to acquire incidence relation between addresses of Bitcoin. In this new method, multi-input transactions, coinbase transactions and change address are studied. It improves the accuracy and comprehensiveness of the relationship between Bitcoin addresses. We verified the comprehensiveness and accuracy of the actual transaction through the Bitcoin addresses we controlled, and the results reach 100%. By using this method, we find 2118 CryptoLocker blackmail addresses.
- (2) We use Louvain [10] clustering algorithm to analyze relation between users. Louvain clustering algorithm that is a community division method based hierarchical clustering can divide transaction addresses closely related to a community so as to find out incidence relation between Bitcoin users. We also improve Louvain clustering algorithm in this paper. In the graph of Bitcoin transactions, we preprocess the leaf nodes and carry out the optimization module of coarse-grained inverse operation in order to increase the community modularity and working efficiency.
- (3) Performance evaluation via extensive experiments also demonstrates that our methods can efficiently trace Bitcoin transaction.

The remainder of this paper is organized as follows: Sect. 2 introduces the method model, definitions, and preliminaries of our work; Sect. 3 gives a concrete description of our measurement methodology; Sect. 4 carries out performance analysis; Sect. 5 introduces relevant work; Sect. 6 concludes this paper.

## 2 Model, Definitions and Preliminaries

### 2.1 Model

We show the system model of transaction tracking in Fig. 1. There are three processes in the model: Data acquisition, Data analysis, and Data presentation. Data acquisition contains Bitcoin transaction data and Bitcoin addresses with disclosed identity in network. Data analysis contains acquisition of transaction relation between Bitcoin addresses and confirmation of relationship between Bitcoin users. Data presentation presents relationship between Bitcoin addresses and users in a visualized way.



**Fig. 1.** System model of transaction tracking.

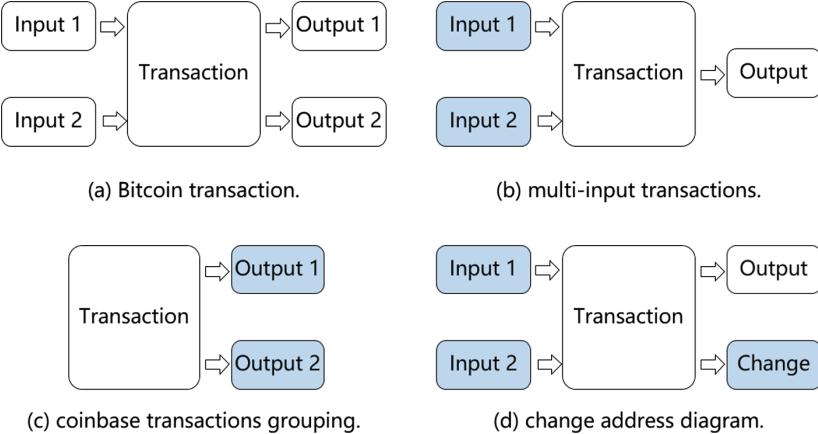
### 2.2 Definitions

**Definition 1 (Address attribution).** The user set is represented as  $U = \{u_1, u_2, \dots, u_n\}$ , the Bitcoin address set is represented as  $A = \{a_1, a_2, \dots, a_n\}$ , and the transaction set is represented as  $T = \{t_1, t_2, \dots, t_n\}$ . The input transaction is represented as  $\text{Input}(t)$ , and the output of the transaction is represented as  $\text{Output}(t)$ .

Bitcoin transaction consists of a set of input addresses, a set of output addresses and change address.

**Definition 2 (Change address).** If a public key  $pk$  meets the following conditions, the  $pk$  is the one-time change address of transaction  $t$ :

- The  $pk$  is only as output of one transaction  $t$ .
- Transaction  $t$  is not a coinbase transaction.
- For  $pk' \in \text{output}(t)$ , no  $pk' \in \text{inputs}(t)$ , i.e. transaction  $t$  is not a transaction of “self-change”.
- No  $pk' \in \text{output}(t)$ , and  $pk' \neq pk$ , but  $pk$  is used as transaction output more than once.

**Fig. 2.** Bitcoin transaction.

**Definition 3 (Transaction Matrix).** When presenting data, we need to convert the user’s transaction data into a matrix. Give an atlas  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ .  $\mathbf{V}$  represents cluster of vertexes of atlas  $\mathbf{G}$  which is transferred from account addresses in Bitcoin transaction network.  $\mathbf{E}$  is the cluster of edges of atlas  $\mathbf{G}$  and is transferred from transaction relation between account addresses in Bitcoin network.

### 2.3 Preliminaries

In this section, we formalize: (i) Bitcoin transaction process revealing incidence relation between Bitcoin addresses, and (ii) Louvain algorithm that shows principle of clustering Bitcoin addresses and incidence relation between users.

**Bitcoin Transaction.** Bitcoin transaction comprises a group of input, output and change address. The input addresses belong to the payer, output addresses belong to the receiver, and change address (optional) is used to store the remaining Bitcoin after transaction, belonging to the payer. The transaction protocol of Bitcoin regulates that the input of a new transaction must be explicit value outputted by previous transaction. Transactions can be divided to single input and multiple outputs, multiple inputs and single output, and multiple inputs and multiple outputs, as shown in Fig. 2(a). In the figure, the input of a new transaction may refer to outputs of multiple transactions previously [2, 3, 9].

**Louvain Algorithm.** Louvain [10] algorithm is based on modularity increment  $\Delta Q$ , and mainly divided to two stages. Firstly, every node is initiated as a community. All nodes in network are traversed ceaselessly, and taken out from

original community. The modularity increment generated by the node's joining in each community is calculated. If the modularity increment is larger than zero, the community with maximum modularity increment shall be selected, and combined with the node. Aforesaid process is repeated until community is not integrated in network [11]. Secondly, a new network will be constructed according to the first layer of community divided, and the weight between new nodes is the weight between original communities. The process of aforesaid stage will be repeated until no community can be combined.

### 3 Measurement Methodology

#### 3.1 Data Acquisition

**Bitcoin Transaction Data.** All transaction data used in this paper is confirmed Bitcoin transactions. We collected them from the blockchain maintained in Bitcoin system, starting from block 1 to block 464283 corresponding to the block creation time from the first one on Jan. 3, 2009 to May 1, 2017 by Bitcoin client with total capacity of 106.87 GB. During this period, a total number of 236242063 Bitcoin transactions have been successfully released and globally confirmed. After acquiring historical transaction, the improved Bitcoin-DatabaseGenerator [12] tool is used to parse the data to acquire data including 284821377 distinct Bitcoin addresses. All the results from this paper are based upon the transactions and addresses from this data set.

**Tracing Data.** This paper aims to study the Bitcoin blackmail event in May 2017. But study shows the three Bitcoin addresses disclosed in the blackmail event are only restricted to receivers accepting Bitcoin sent by victims, and no further transaction has been found yet. Therefore, this paper acquires and verifies experimental data by example of similar Bitcoin blackmail virus CryptoLocker occurred in 2013 to verify research designs. Program Scrapy web spider and get disclosed Bitcoin addresses in relevant forum to trace flow of Bitcoin. By Scrapy, 5 CryptoLocker blackmail addresses are acquired.

#### 3.2 Data Analysis

**Confirmation of Bitcoin Addresses' Incidence Relation.** In Bitcoin transaction network, the addresses are connected by transaction activities. Thus, it can be confirmed that two addresses connected are of certain incidence relation; and the source and flow direction of Bitcoin can be known according to characteristics of Bitcoin transaction protocol. We propose a new heuristic method with three rules to acquire incidence relation between addresses of Bitcoin. The method can confirm which Bitcoin addresses belong to the same user, and it can be concluded and described as:

- ***multi-input transactions grouping***

If two or more addresses are inputs of the same transaction, they are controlled by the same user; for instance, for any transaction  $t$ , all  $pk \in \text{inputs}(t)$  are controlled by the same user, as shown in Fig. 2(b).

- ***coinbase transactions grouping***

If two or more addresses are outputs of the same coinbase transaction, they are controlled by the same user; for instance, for any coinbase transaction, all  $pk \in \text{output}(t)$  are controlled by the same user, as shown in Fig. 2(c).

- ***change address guessing***

The one-time change address and transaction input addresses are controlled by the same user; for instance, for any transaction  $t$ , the controller of  $\text{input}(t)$  controls the one-time change address  $pk \in \text{output}(t)$ , as shown in Fig. 2(d).

**Confirmation of Bitcoin Users' Incidence Relation.** During transaction, every user can participate in transaction by multiple Bitcoin addresses. As previously mentioned, different Bitcoin addresses of the same user are confirmed. However, for the Bitcoin blackmail event just occurred in May 2017, maybe criminal gang comprises many people and everyone has multiple Bitcoin address. After the blackmail, criminals gathered the Bitcoin blackmailed to an account of a higher-level member. In this section, we set up a data set of all blackmail addresses which we found to determine the relationship between users. The strategy of this paper is to adopt Louvain algorithm to confirm Bitcoin users' incidence relation and improve it. We improve time complexity and optimal modular based on the enhanced Louvain algorithm proposed by Gach and Hao [13], and it can be concluded and described as:

- ***Node pretreatment***

In Bitcoin transaction network, account address participating in transaction is node of the network; transactions between accounts are edges connecting nodes. For address  $i$  and address  $j$ , if  $i$  is the only connection of  $j$ , address  $j$  will surely be divided to the same community with address  $i$ , which can be proved:

If as assumed before, node  $i$  and node  $j$  belong to the same community, then:

$$Q_{i \rightarrow C(j)} = \sum(A_{ij}) - a_j^2$$

Increment of corresponding modularity:

$$\Delta Q_{i \rightarrow C(j)} = \frac{\sum_{i,j}}{2m^2} (2m - k_j)$$

If node  $i$  and node  $j$  are not in the same community,  $Q_{i \rightarrow C(j)} \leq 0$ , then  $2m < k_j$ . Thus, if node  $i$  is the only connection of node  $j$ , node  $j$  will surely be divided to the same community with node  $i$ . If node  $j$  is classified before community division, the modularity  $Q$  computation of certain nodes can be reduced so as to improve efficiency of community division.

- ***optimized modularity***

In the enhanced Louvain algorithm proposed by Gach and Hao [13]. During coarseness inverse operation optimization, the attribution of nodes in the community connecting with outside will be confirmed again by K-medoids

algorithm [14]. Take the node and the node most closely connecting to node in the community as mass points to calculate community attribution of the node. Since number of  $V_i$  and  $V_j$  is limited, K-medoids algorithm implementation efficiency is high, which can save time to recalculate several  $\Delta Q$  of every  $V_i$ , so as to improve efficiency.

### 3.3 Data Presentation

To better understand the relation of Bitcoin addresses and users, we use Gephi [15], an open source software for network visualization and exploration, to visualize outgoing transactions from Bitcoin transaction data.

**Presentation of Addresses Transaction Relation.** Take a CryptoLocker blackmail address to carry out test. We use “multi-input transactions grouping”, “coinbase transactions grouping”, and “change address guessing” to cluster addresses, and the heuristic method iterate tenth. The addresses incidence relation graph is acquired. The addresses belong to the same user. Take these addresses as vertexes, and transactions between addresses as edges; save after converting to graph, and output visually.

**Presentation of Users Incidence Relation.** As previously mentioned, we set up a data set of all blackmail addresses which we found. Take these addresses as vertexes, and transaction between addresses as edges. By improved Louvain algorithm, we mark the addresses belonging to the same user as the same color, and we distinguish 17 distinct sub-communities in the CryptoLocker blackmail addresses network. We see that the ransom balances from all addresses within a community are transferred to a single aggregate address at the center.

## 4 Performance Analysis

The scheme is implemented in Python language. The database of Bitcoin blockchain data storage is SQL Server. The experimental machine with 2.40 GHz, Intel i5-2430M CPU and 8 GB RAM.

### 4.1 Comprehensiveness and Accuracy

We have carried out 45 transactions on the two Bitcoin addresses which we have controlled. The actual transaction addresses involving our control are 126 and 158 respectively, and the experimental results are in good agreement with the actual data, as shown in Table 1. The results indicate that the heuristic method of this paper has a good comprehensiveness and accuracy.

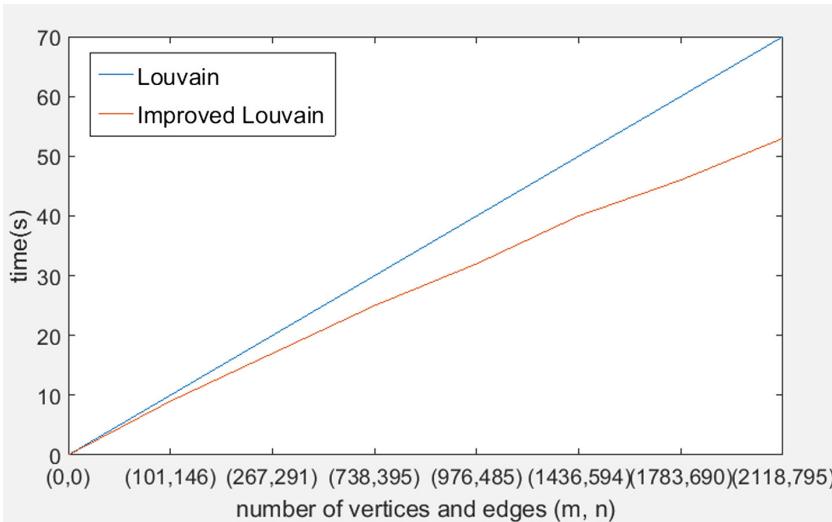
We use the CryptoLocker blackmail addresses which are acquired By Scrapy. Meanwhile, according to burst time of the virus and amount of Bitcoin transacted, 2118 victim addresses, 795 transactions and 1128.40 Bitcoin value are inquired from historical Bitcoin transaction data by our heuristic method.

**Table 1.** Experimental results of test addresses

Test address sample	Number of associated addresses	Comprehensive rate	Accuracy rate
1***P	126	100%	100%
1***y	158	100%	100%

## 4.2 Efficiency

In the Bitcoin transaction network, the address represents the vertex, and the transaction between the addresses represents the edge. As mentioned earlier, there are 2118 vertices and 795 edges. In order to verify the efficiency of the algorithm, the Bitcoin transaction data is processed by Louvain algorithm and improved Louvain algorithm respectively, and the data volume is gradually increased. Figure 3 shows a comparison of accumulated average runtime between Louvain algorithm and improved Louvain algorithm with different data volume. The results show that improved algorithm reduces runtime by about 4.533% compared with the original algorithm.

**Fig. 3.** Runtime comparison between Louvain and improved Louvain.

## 5 Related Work

In recent years, security and privacy issues have been a hot topic of research [16–25, 27, 28], and Bitcoin privacy issues are a key attention. The related work can be classified into two categories: clustering analysis based on incidence relationship of Bitcoin address and Louvain community algorithm clustering analysis.

### 5.1 Clustering Analysis Based on Incidence Relationship of Bitcoin Addresses

Large number of researches show inherent flaws of Bitcoin system in privacy. Reid and Harrigan [2] found incidence relationship between transaction addresses by studying Bitcoin transaction, generated transaction network and user network, analyzed quantity, amount and related address of transaction incurred by addresses that Wiki Leaks disclosed so as to find out flow direction of Bitcoin. Meiklejohn et al. [3] heuristically studied cluster of multi-input transaction address and change address. Ron and Shamir [4] generated user map pursuant to incidence relationship of transactions, carried out in-depth study on the largest Bitcoin transaction in history, and concluded based on the data that Bitcoin system has large amount of hoarding behavior, and most capital is not circulating. Androulaki et al. [5] carried out Bitcoin privacy test by actual Bitcoin system and simulating Bitcoin system in university; deeply studied change address; 40% users were participated in the test, and user data could effectively realize deanonymisation of Bitcoin by clustering technology with accuracy of 80%. Zhao [6] studied clustering of multi-input transaction, coinbase transaction and change address, and found out flow direction of Bitcoin. Spagnuolo et al. [7] used heuristic method to realize clustering of multi-input transaction and change address. Monaco [8] identified user by measuring biological characteristics of user identity according to time sequence of transaction sample during a period of time. Liao et al. [9] studied flow of blackmail software ransom, and traced blackmailing addresses by classifying addresses receiving ransom by clustering technology.

### 5.2 Louvain Community Algorithm

Bitcoin transaction data is vast, and relationship is complicated. The community division method based hierarchical clustering can divide transaction addresses closely related to a community so as to find out incidence relation between Bitcoin addresses. Blondel et al. [10] proposed Louvain algorithm, which divides community by modularity calculation, and can rapidly process big data. Gach and Hao [13] proposed an enhanced Louvain algorithm, and adopted multi-level method to maximize module. De Meo et al. [26] optimizes modularization ideas of Louvain algorithm. The optimal program is to realize maximum network module by computing route from central point so as to improve operating efficiency.

## 6 Conclusion

This paper clusters incidence relation between Bitcoin addresses by a new heuristic method, and further confirms incidence relation between users by improved Louvain algorithm. However, the heuristic method mentioned in this paper may generate certain error for judgment on change address. Louvain algorithm needs to be further improved for efficiency implementation. Different iteration frequencies of the two methods may lead in different quantities. But the larger the

iteration frequency is, and the lower efficiency will be. In the future, studies can be carried out around those problems.

**Acknowledgements.** This work was supported in part by the National Science Foundation of China under Grant 61602039, in part by the Beijing Natural Science Foundation under Grant 4164098, in part by CCF-Venustech Open Research Fund, in part by BIT-UMF research and development fund, and in part by Education and Teaching Reform Project of China University of Political Science and Law under Grant 1000/10717130.

## References

1. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. Consulted (2008)
2. Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system, pp. 1318–1326 (2011)
3. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., Mccoy, D., Voelker, G.M., Savage, S.: A fistful of Bitcoins: characterizing payments among men with no names. In: Conference on Internet Measurement Conference, pp. 127–140. ACM (2013)
4. Ron, D., Shamir, A.: Quantitative analysis of the full Bitcoin transaction graph. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 6–24. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-39884-1\\_2](https://doi.org/10.1007/978-3-642-39884-1_2)
5. Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in Bitcoin. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 34–51. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-39884-1\\_4](https://doi.org/10.1007/978-3-642-39884-1_4)
6. Zhao, C.: Graph-based forensic investigation of Bitcoin transactions (2014)
7. Spagnuolo, M., Maggi, F., Zanero, S.: BitIodine: extracting intelligence from the bitcoin network. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 457–468. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-662-45472-5\\_29](https://doi.org/10.1007/978-3-662-45472-5_29)
8. Monaco, J.V.: Identifying Bitcoin users by transaction behavior. In: SPIE DSS (2015)
9. Liao, K., Zhao, Z., Doupe, A., Ahn, G.J.: Behind closed doors: measurement and analysis of CryptoLocker ransoms in Bitcoin. In: Electronic Crime Research, pp. 1–13. IEEE (2016)
10. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *J. Stat. Mech. Theor. Exp.* **30**, 155–168 (2008)
11. (U) Bitcoin Virtual Currency: Unique Features Present Distinct Challenges for Deterring Illicit Activity (2011)
12. <https://github.com/ladimolnar/BitcoinDatabaseGenerator/releases>
13. Gach, O., Hao, J.-K.: Improving the Louvain algorithm for community detection with modularity maximization. In: Legrand, P., Corsini, M.-M., Hao, J.-K., Monmarché, N., Lutton, E., Schoenauer, M. (eds.) EA 2013. LNCS, vol. 8752, pp. 145–156. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-11683-9\\_12](https://doi.org/10.1007/978-3-319-11683-9_12)
14. Park, H.S., Jun, C.H.: A simple and fast algorithm for K-medoids clustering. *Expert Syst. Appl.* **36**(2), 3336–3341 (2009)
15. Gephi. <https://gephi.org/>
16. Shen, M., Ma, B., Zhu, L., Mijumbi, R., Du, X., Hu, J.: Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection. *IEEE Trans. Inf. Forensics Secur.* **13**(4), 940–953 (2018)

17. Du, X., Shayman, M., Rozenblit, M.: Implementation and performance analysis of SNMP on a TLS/TCP base. In: IEEE/IFIP International Symposium on Integrated Network Management Proceedings IEEE, pp. 453–466 (2001)
18. Du, X., Wu, D.: Adaptive cell relay routing protocol for mobile ad hoc networks. *IEEE Trans. Veh. Technol.* **55**(1), 278–285 (2006)
19. Zhang, M., Nygard, K.E., Guizani, S.: Self-healing sensor networks with distributed decision making. *Int. J. Sens. Netw.* **2**(5/6), 289–298 (2007)
20. Du, X., et al.: An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Netw.* **5**(1), 24–34 (2007)
21. Du, X., Chen, H.H.: Security in wireless sensor networks. *Wirel. Commun. IEEE* **15**(4), 60–66 (2008)
22. Xiao, Y., Chen, H.H., Du, X., et al.: Stream-based cipher feedback mode in wireless error channel. *IEEE Trans. Wirel. Commun.* **8**(2), 622–626 (2009)
23. Du, X., Guizani, M., Xiao, Y., Chen, H.H.: A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. *IEEE Trans. Wirel. Commun.* **8**(3), 1223–1229 (2009)
24. Yao, X., Han, X., Du, X., Zhou, X.: A lightweight multicast authentication mechanism for small scale IoT applications. *IEEE Sens. J.* **13**(10), 3693–3701 (2013)
25. Liang, S., Du, X.: Permission-combination-based scheme for Android mobile malware detection. In: IEEE International Conference on Communications, pp. 2301–2306. IEEE (2014)
26. De Meo, P., Ferrara, E., Fiumara, G., Provetti, A.: Generalized Louvain method for community detection in large networks. In: International Conference on Intelligent Systems Design and Applications, pp. 88–93. IEEE (2012)
27. Fahad, A., Alshatri, N., Tari, Z., et al.: A survey of clustering algorithms for big data: taxonomy and empirical analysis. *IEEE Trans. Emerg. Top. Comput.* **2**(3), 267–279 (2014)
28. Almalawi, A.M., Fahad, A., Tari, Z., Cheema, M.A., Khalil, I.: kNNVWC: an efficient k-nearest neighbors approach based on various-widths clustering. *IEEE Trans. Knowl. Data Eng.* **28**(1), 68–81 (2016)



# Cryptanalysis of Salsa and ChaCha: Revisited

Kakumani K. C. Deepthi<sup>(✉)</sup> and Kunwar Singh

Computer Science and Engineering Department,  
National Institute of Technology, Tiruchirappalli, India  
[{406115002,kunwar}@nitt.edu](mailto:{406115002,kunwar}@nitt.edu)

**Abstract.** Stream cipher is one of the basic cryptographic primitives that provide the confidentiality of communication through insecure channel. EU ECRYPT network has organized a project for identifying new stream suitable for widespread adoption where the ciphers can provide a more security levels. Finally the result of the project has identified new stream ciphers referred as eSTREAM. Salsa20 is one of the eSTREAM cipher built on a pseudorandom function. In this paper our contribution is two phases. First phase have two parts. In WCC 2015, Maitra et al. [9] explained characterization of valid states by reversing one round of Salsa20. In first part, we have revisited the Maitra et al. [9] characterization of valid states by reversing one round of Salsa20. We found there is a mistake in one bit change in 8<sup>th</sup> and 9<sup>th</sup> word in first round will result in valid initial state. In second part, Maitra et al. [9] as mentioned that it would be an interesting combinatorial problem to characterize all such states. We have characterized nine more values which lead to valid initial states. The combinations  $(s_4, s_7)$ ,  $(s_2, s_3)$ ,  $(s_{13}, s_{14})$ ,  $(s_1, s_6)$ ,  $(s_1, s_{11})$ ,  $(s_1, s_{12})$ ,  $(s_6, s_{11})$ ,  $(s_6, s_{12})$  and  $(s_{11}, s_{12})$  which characterized as valid states.

In second phase, FSE 2008 Aumasson et al. [1] attacked 128-key bit of Salsa20/7 within  $2^{111}$  time and ChaCha6 in within  $2^{107}$  time. After this with best of our knowledge there does not exist any improvement on this attack. In this paper we have attacked 128-key bit of Salsa20/7 within  $2^{107}$  time and ChaCha6 within  $2^{102}$  time. Maitra [8] improved the attack on Salsa20/8 and ChaCha7 by choosing proper IVs corresponding to the 256-key bit. Applying the same concept we have attacked 128-key bit of Salsa20/7 within time  $2^{104}$  and ChaCha7 within time  $2^{101}$ .

**Keywords:** Stream cipher · eSTREAM · Salsa · ChaCha  
Non-randomness · Quarterround · Reverseround · Valid states  
Probabilistic neutral bit (PNB) · ARX cipher

## 1 Introduction

Stream cipher is one of the basic cryptographic primitives that provide the confidentiality of communication through insecure channel. It produces a long pseudorandom sequence called keystream from short random string called secret key

(seed). Here encryption of message is carried out bit by bit which can be achieved XORing the keystream to the message. Receiver regenerates the keystream from shared secret key (seed) then decryption is achieved XORing the key stream to the ciphertext. Single secret key is used to encrypt two different messages which is vulnerable to some kind of attack. So, for each and every message there should be different key which cannot be considered practically.

In the modern time, a pseudo-random generator used in stream cipher which depends on a secret key (seed) and Initialization Vector (IV). Here IV does not need to be kept secret and it must change for every encryption session, which it is used as randomizer. The IV is communicated to the receiver publicly (TRIVIUM), or could be combination of this publicly communicated value and a counter value generated at both ends (SALSA 20/12). This gives the flexibility that same key can be used to different messages. Security model of these stream cipher captures the idea that for distinct values of IV with same key, the output of a pseudo-random generator should appear uniform random to an adversary with polynomial bounded computational power.

EU ECRYPT network has organized a project for identifying new stream suitable for widespread adoption where the ciphers can provide a more security levels. Finally the result of the project has identified new stream ciphers referred as eSTREAM. Salsa20 [3] is the one of eSTREAM which provides much better speed-security profile. Salsa20 offers a very simple, clean, and scalable design developed by Daniel J. Bernstein. It supports 128-bit and 256-bit keys in a very natural way. The simplicity and scalability of the algorithm has given more importance in cryptanalytic area. ChaCha [2] is family of stream ciphers, a variant of Salsa published by Daniel J. Bernstein. ChaCha has better diffusion per round and increasing resistance to cryptanalysis.

**Related Work:** In SASC 2005, Crowley [5] has reported first attack in Salsa20 and won Bernstein's US\$1000 prize amount for "most interesting Salsa20 cryptanalysis". He presented an attack on Salsa20 reduced to five of its twenty rounds and is based on truncated differentials. Truncated differential cryptanalysis is a generalization of differential cryptanalysis, an attack against block ciphers. In INDOCRYPT 2006, Fisher et al. [7] has reported in Salsa20/6 and Salsa20/7 an attack and presented a key recovery attack on six rounds and observe non-randomness after seven rounds. In SASC 2007, Tsunoo et al. [12] reported that there is a significant bias in the differential probability for Salsa20's 4<sup>th</sup> round internal state. In FSE 2008, Aumasson et al. [1] have reported an attack which makes use of the new concept of probabilistic neutral key bits (PNB). PNB is the process of identifying a large subset of key bits which can be replaced by fixed bits so that detectable bias after approximate backwards computation is still significant. This attack was further improved by Shi et al. [11] using the concept of Column Chaining Distinguisher (CCD). By choosing IVs corresponding to the keys, Maitra [8] improved the attack on Salsa20/8. Further the attacks on Salsa and ChaCha were improved by Choudhuri and Maitra [4] and Sabyasachi and Sarkar [6]. All these attacks are on 256 version of Salsa20/8. In WCC 2015, Maitra et al. [9] provided interpretation based on Fisher's 2006

result. Maitra et al. [9] included the key bits in the PNB set by providing less probability for distinguishing. In this way they have considered and obtained ( $36 + 5 = 41$ ) PNBs by key recovery attack with key search complexity of  $2^{247.2}$ . Maitra et al. [9] characterization of valid initial state by reversing one round of Salsa20, which helps in obtaining sharper bias value. It is found that change in some bit position after one round can obtain the initial value state by reversing one round back. Maitra et al. [9] given that one bit change in 8<sup>th</sup> and 9<sup>th</sup> word in first round will result in valid initial state as follows.

$$\begin{bmatrix} 0 & 0 & 0x80000000 & 0 \\ 0 & 0 & 0x80001000 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0x???80040 & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0x80000000 & 0x80000000 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (1)$$

**Our Contribution:** Our contribution is of two phases. First phase have two parts. In first part, we have revisited the Maitra et al. [9] characterization of valid states by reversing one round of Salsa20. We found there is a mistake in Eq. (1) in which one bit change of 8<sup>th</sup> and 9<sup>th</sup> word in first round will result in valid initial state. Instead of 0x???80040, it should be as follows.

$$\begin{bmatrix} 0 & 0 & 0x80000000 & 0 \\ 0 & 0 & 0x80001000 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \textbf{0x}???800?? & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0x80000000 & 0x80000000 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

In second part, Maitra et al. [9], as mentioned that it would be an interesting combinatorial problem to characterize all such states. We have characterized nine more values which lead to valid initial states. The combinations  $(s_4, s_7)$ ,  $(s_2, s_3)$ ,  $(s_{13}, s_{14})$ ,  $(s_1, s_6)$ ,  $(s_1, s_{11})$ ,  $(s_1, s_{12})$ ,  $(s_6, s_{11})$ ,  $(s_6, s_{12})$  and  $(s_{11}, s_{12})$  which characterized as valid states.

In second phase, FSE 2008 Aumasson et al. [1] attacked 128-key bit of Salsa20/7 within  $2^{111}$  time and ChaCha6 in within  $2^{107}$  time. After this with best of our knowledge there does not exist any improvement on this attack. In this paper we have attacked 128-key bit of Salsa20/7 within  $2^{107}$  time and ChaCha6 within  $2^{102}$  time. Maitra [8] improved the attack on Salsa20/8 and ChaCha7 by choosing proper IVs corresponding to the 256-key bit. Applying the same concept we have attacked 128-key bit of Salsa20/7 within time  $2^{104}$  and ChaCha7 within time  $2^{101}$ .

**Paper Outline:** Our paper is organized as follows. In Sect. 2, with preliminaries we describe Salsa20 specification, ChaCha specification, differential analysis, PNBs and estimation of the complexity. Section 3, Maitra et al. [9] work characterization of valid state. In Sect. 4, we describe our work. In Sect. 5, we give conclusion and related open problems.

## 2 Preliminaries

### 2.1 Salsa20 Specification

Salsa20 is built on a pseudorandom function based on Add-Rotate-XOR (ARX) operations. It carries 32-bit addition, left rotation and bitwise addition as XOR. Salsa20 stream cipher considers 16 words, each of 32-bit. Basic structure is as follows.

$$S = \begin{bmatrix} s_0 & s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 & s_7 \\ s_8 & s_9 & s_{10} & s_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{bmatrix} = \begin{bmatrix} c_0 & k_0 & k_1 & k_2 \\ k_3 & c_1 & v_0 & v_1 \\ t_0 & t_1 & c_2 & k_4 \\ k_5 & k_6 & k_7 & c_3 \end{bmatrix}$$

The first matrix with  $(s_0, s_1, s_2, \dots, s_{15})$  represents the words of initial states and in second matrix where  $c_0, c_1, c_2, c_3$  are the predefined constants,  $k_0, k_1, \dots, k_7$  represents key of 256-bit,  $IV = (t_0, t_1, v_0, v_1)$  represents  $t_0, t_1$  are the 64-bit counter and  $v_0, v_1$  are the 64-bit nonce. Constant values may be changing based on the key value. If we are considering 256-bit key then we can call it as 256-bit Salsa else if we are considering 128-bit key then it is 128-bit Salsa.

The main operation in Salsa20 is carried out by a nonlinear operation called quarterround function. One quarterround function is the four ARX rounds. One ARX round is the one addition (A) plus one cyclic left rotation (R) plus one XOR (X). Quarterround ( $w, x, y, z$ ) is defined as follows.

$$\begin{aligned} x &= x \oplus ((w + z) \lll 7) \\ y &= y \oplus ((x + w) \lll 9) \\ z &= z \oplus ((y + x) \lll 13) \\ w &= w \oplus ((z + y) \lll 18) \end{aligned}$$

The addition (A) is carried out by two words and result is divided modulo by  $2^{32}$ , cyclic left rotation (R) is carried out for each bit in the given word that is the leftmost bits move to the rightmost positions and exclusive-or (X) is carried out by sum of the two words with carries suppressed.

The rounds in Salsa20 can be performed based on column and row of the matrix as the initial states are considered. So performing round in row of matrix is called as rowround and column of matrix as columnround. If it is a columnround then one columnround is the four quarterrounds, one on each of the four columns of the initial state matrix. If it is rowround then one rowround is the four quarterrounds, one on each of the four rows of the initial state matrix.

Suppose if  $S$  is a 16-word input with values  $(s_0, s_1, \dots, s_{15})$  then the rowround( $S$ ) and columnround( $S$ ) is as follows

Rowround:

$$\begin{aligned} \text{quarterround } &(s_0, s_1, s_2, s_3) \\ \text{quarterround } &(s_5, s_6, s_7, s_4) \\ \text{quarterround } &(s_{10}, s_{11}, s_8, s_9) \\ \text{quarterround } &(s_{15}, s_{12}, s_{13}, s_{14}) \end{aligned}$$

Columnround:

$$\begin{aligned} \text{quarterround } &(s_0, s_4, s_8, s_{12}) \\ \text{quarterround } &(s_5, s_9, s_{13}, s_1) \\ \text{quarterround } &(s_{10}, s_{14}, s_2, s_6) \\ \text{quarterround } &(s_{15}, s_3, s_7, s_{11}) \end{aligned}$$

The **one round [9,10] of the Salsa20** can also be considered by one columnround and one transpose of the initial state matrix (so 12 rounds can be considered as one columnround and one transpose of 12 times). In this paper we

will be considering Salsa20 as 20 rounds in which one round as one columnround and one transpose.

Let  $S^{(0)}$  be the initial state  $S$  and  $S^{(r)}$  be  $r$  rounds applied on the initial state  $S$ . So that after  $R$  rounds it becomes  $S^{(R)}$ . Then the keystream for 512 bits is obtained as  $Z = S + S^{(R)}$ .

**Reversing One Round:** Maitra et al. [9] given that Salsa20 round can be reversible as the state-transition operations are reversible. If  $S^{(r+1)}$ , then  $S^{(r)} = \text{reverseround}(S^{(r+1)})$ . Where reverseround is the inverse of the round and consists of first transposing the state and then applying the inverse of quarterround for each column by reverseround. Reverseround ( $w, x, y, z$ ) is as follows.

$$w = w \oplus ((z + y) \lll 18)$$

$$z = z \oplus ((y + x) \lll 13)$$

$$y = y \oplus ((x + w) \lll 9)$$

$$x = x \oplus ((w + z) \lll 7)$$

Reverseround works as follows.

1. Consider initial matrix after first round be  $S_1$  and perform transpose be  $S^T$ .

$$S_1 = \begin{bmatrix} s'_0 & s'_1 & s'_2 & s'_3 \\ s'_4 & s'_5 & s'_6 & s'_7 \\ s'_8 & s'_9 & s'_{10} & s'_{11} \\ s'_{12} & s'_{13} & s'_{14} & s'_{15} \end{bmatrix} \quad S^T = \begin{bmatrix} s'_0 & s'_4 & s'_8 & s'_{12} \\ s'_1 & s'_5 & s'_9 & s'_{13} \\ s'_2 & s'_6 & s'_{10} & s'_{14} \\ s'_3 & s'_7 & s'_{11} & s'_{15} \end{bmatrix}$$

2. Perform reverseround by column wise for each column individually. Where reverseround ( $w, x, y, z$ ) is equal to  $[s'_0, s'_1, s'_2, s'_3]$ ,  $[s'_5, s'_6, s'_7, s'_4]$ ,  $[s'_{10}, s'_{11}, s'_{12}, s'_{13}]$  and  $[s'_{15}, s'_{12}, s'_{13}, s'_{14}]$ .

## 2.2 ChaCha Specification

ChaCha is a family of stream cipher, a variant of Salsa. It is similar to Salsa is of 16 words, each 32-bit. Basic structure is as follows

$$S = \begin{bmatrix} s_0 & s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 & s_7 \\ s_8 & s_9 & s_{10} & s_{11} \\ s_{12} & s_{13} & s_{14} & s_{15} \end{bmatrix} = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ t_0 & v_0 & v_1 & v_2 \end{bmatrix}$$

The first matrix with  $(s_0, s_1, s_2, \dots, s_{15})$  represents the words of initial states and in second matrix where  $c_0, c_1, c_2, c_3$  are the predefined constants,  $k_0, k_1, \dots, k_7$  represents key of 256-bit,  $IV = (t_0, t_1, v_0, v_1)$  represents  $t_0, t_1$  are the 64-bit counter and  $v_0, v_1$  are the 64-bit nonce. In ChaCha nonlinear operations are slightly different from Salsa and are as follows

$$w = w + x; z = z \oplus w; z = z \lll 16;$$

$$y = y + z; x = x \oplus y; x = x \lll 12;$$

$$w = w + x; z = z \oplus w; z = z \lll 8;$$

$$y = y + z; x = x \oplus y; x = x \lll 7;$$

As in Salsa columnround and rowround are considered, but here in ChaCha columnround and diagonalround are considered. It is considered as for odd rounds first columnround is applied and for even rounds first diagonalround is applied. Columnround and diagonalrounds are as follows

Columnround:	Diagonalround:
quarterround ( $s_0, s_4, s_8, s_{12}$ )	quarterround ( $s_0, s_5, s_{10}, s_{15}$ )
quarterround ( $s_1, s_5, s_9, s_{13}$ )	quarterround ( $s_1, s_6, s_{11}, s_{12}$ )
quarterround ( $s_2, s_6, s_{10}, s_{14}$ )	quarterround ( $s_2, s_7, s_8, s_{13}$ )
quarterround ( $s_3, s_7, s_{11}, s_{15}$ )	quarterround ( $s_3, s_4, s_9, s_{14}$ )

Like in Salsa, in ChaCha also reverseround is the inverse of round and defined as follows

$$\begin{aligned} x &= x \ggg 7; x = x \oplus y; y = y - z; \\ z &= z \ggg 8; z = z \oplus w; w = w - x; \\ x &= x \ggg 12; x = x \oplus y; y = y - z; \\ z &= z \ggg 16; z = z \oplus w; w = w - x; \end{aligned}$$

### 2.3 Differential Analysis

The differential attack is that some small differences in input states have a perceptible chance in producing small differences after the first round of computation, the second round of computation, etc. The behavior of the differential is heavily key-dependent. There are many unbalanced bits in the states of Salsa20 after four rounds.

Let  $s_i$  is the  $i^{th}$  word of the matrix S and  $s_{i,j}$  is the  $j^{th}$  least significant bit of  $s_i$ . Suppose if we are having two states  $S^{(r)}$  and  $S'^{(r)}$ , after r rounds it can be denoted as  $\Delta_i^{(r)} = s_i^{(r)} \oplus s_i'^{(r)}$ . Thus

$$\Delta^{(r)} = S^{(r)} \oplus S'^{(r)} = \begin{bmatrix} \Delta_0^{(r)} & \Delta_1^{(r)} & \Delta_2^{(r)} & \Delta_3^{(r)} \\ \Delta_4^{(r)} & \Delta_5^{(r)} & \Delta_6^{(r)} & \Delta_7^{(r)} \\ \Delta_8^{(r)} & \Delta_9^{(r)} & \Delta_{10}^{(r)} & \Delta_{11}^{(r)} \\ \Delta_{12}^{(r)} & \Delta_{13}^{(r)} & \Delta_{14}^{(r)} & \Delta_{15}^{(r)} \end{bmatrix}$$

After performing few rounds biases can be obtained. For that take Input Differential (ID) at the initial state and try to obtain some biases value corresponding to combinations of some output bits as Output Differential (OD).

Let  $S^{(1)}$  and  $S'^{(1)}$  be the two initial states that differ in a few places. That is probability is different at the  $q^{th}$  bit of the  $p^{th}$  word and they are same at all the other bits of the complete state or differ at the  $j^{th}$  bit of the  $i^{th}$  word and they are same at all the other bits of the complete state is the amount of bias and is computed as  $Pr(\Delta_{p,q}^{(r)} = 1 | \Delta_{i,j}^{(0)} = 1) = \frac{1}{2}(1 + \epsilon_d)$ , where  $\epsilon_d$  is a measure of the bias. Here the probability is estimated for fixed key and by all the possible choices of nonces and counters, other than the constraints imposed due to the input differences on the nonces or counters.

## 2.4 Probabilistic Neutral Bits (PNBs)

In FSE 2008, Aumasson et al. [1] have reported an attack which makes use of the new concept of probabilistic neutral key bits (PNB) for probabilistic detection of a truncated differential. In 2015, Maitra et al. [9] revisited the concept and explained the concept in simple manner. PNB is the process of identifying a large subset of key bits which can be replaced by fixed bits so that detectable bias after approximate backwards computation is still significant.

**Setting Up:** Consider  $S$  and  $S'$  are the two initial states change in  $j$ -th bit of  $i$ -th word for the given input differential  $\Delta_{i,j}^{(0)}$ , by executing Salsa algorithm for  $r$  rounds i.e.,  $r < R$  observed a high bias  $\epsilon_d$  in the output differential  $\Delta_{p,q}^{(r)}$  by randomly choosing keys, nonces and counters. Bias is estimated by  $Pr(\Delta_{p,q}^{(r)} = 1 | \Delta_{i,j}^{(0)} = 1) = \frac{1}{2}(1 + \epsilon_d)$ , where  $\epsilon_d$  is a measure of the bias.

**PNB Concept:** Now execute Salsa algorithm for  $R$  rounds then  $Z = S + S^{(R)}$  and  $Z' = S' + S'^{(R)}$  are two keystream blocks. By complementing particular key bit position  $k$  in  $S$  and  $S'$  yields to the states  $\bar{S}$  and  $\bar{S}'$ . After executing reverse  $R - r$  rounds by  $Z - \bar{S}$  and  $Z' - \bar{S}'$  results  $\bar{Y}$  and  $\bar{Y}'$ . Let the difference is considered as  $\Gamma_{p,q} = Y_{p,q} \oplus Y'_{p,q}$ . Now compare this difference with the previous calculated after  $r$  rounds of the Salsa. The bias  $Pr(\Delta_{p,q}^{(r)} = \Gamma_{p,q} | \Delta_{i,j}^{(0)} = 1)$  is high then the key bit  $k$  is probabilistic neutral bit (PNB). The neutrality measure of the key bit is  $Pr(\Delta_{p,q}^{(r)} = \Gamma_{p,q} | \Delta_{i,j}^{(0)} = 1) = \frac{1}{2}(1 + \gamma_k)$ .

**To Obtain PNBs:** We have experimented for  $2^{24}$  samples (randomly choosing nonce and counter) corresponding to each key bit. We have repeated this process for all 128-key bit to obtain the PNBs. A threshold probability  $\frac{1}{2}(1 + \gamma)$  is chosen to filter PNBs i.e., if  $\gamma_k \geq \gamma$ , then key bit  $k$  is included in the set of the PNBs. So, the whole set of key bits are divided into PNBs and non-PNBs set. Let  $n$  be the set of PNBs and  $m$  be set of non-PNBs, then the size of whole set is ( $m + n = 128$ ).

**Attack Idea:** Main idea of attack considers search over the key bits which are non-PNBs without knowing the correct values of PNBs. The correct key values have been assigned to the  $m$  non-PNBs key bits and random binary values are assigned to the  $n$  PNBs key bits in both  $S$  and  $S'$  yields to  $\hat{S}$  and  $\hat{S}'$ . Then compute  $Z - \hat{S}$  and  $Z' - \hat{S}'$  and apply  $R - r$  rounds on both of them to result  $\hat{Y}$  and  $\hat{Y}'$ . Forward Salsa by  $r$  rounds results  $S^r$  and  $S'^r$  respectively. The difference is  $\hat{\Gamma}_{p,q} = \hat{Y}_{p,q} \oplus \hat{Y}'_{p,q}$ . Probability is  $Pr(\hat{\Gamma}_{p,q} = 1 | \Delta_{p,q}^{(r)} = 1) = \frac{1}{2}(1 + \epsilon_a)$  by which PNBs can be identified. The bias after  $R$  rounds is  $Pr(\hat{\Gamma}_{p,q} = 0) = \frac{1}{2}(1 + \epsilon)$ . Where  $\epsilon \approx \epsilon_a \cdot \epsilon_d$ . One can make exhaustive search over all possible keys ( $2^{128}$ ). Most probable key is that key which has high bias  $\epsilon^*$  value. Where  $\epsilon^*$  median of the all  $\epsilon$ 's.

## 2.5 Estimation of Complexity

By revisiting FSE 2008, Aumasson et al. [1] the complexity of the attack is given by

$$2^m(N + 2^n P_{fa}) = 2^m N + 2^{256-\alpha}.$$

Here m and n are the number of non-PNBs and PNBs. N is the probabilities of required number of samples and by Neyman-Pearson decision theory

$$N \approx \left( \frac{\sqrt{\alpha \log 4} + \sqrt[3]{1 - \epsilon^*}^2}{\epsilon^{*2}} \right)^2$$

where  $P_{nd} = 1.3 \times 10^{-3}$  and  $P_{fa} = 2^{-\alpha}$ .

## 3 Maitra et al. [9] Characterization of Valid State

Maitra et al. [9] has found that one bit change in 8<sup>th</sup> and 9<sup>th</sup> word in first round will result in valid initial state as follows.

$$\begin{bmatrix} 0 & 0 & 0x80000000 & 0 \\ 0 & 0 & 0x80001000 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0x???80040 & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0x80000000 & 0x80000000 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Here we have gone through detail calculation of the 8<sup>th</sup> and 9<sup>th</sup> word we observed that there is a mistake in Maitra et al. [9] and is as follows.

$$\begin{bmatrix} 0 & 0 & 0x80000000 & 0 \\ 0 & 0 & 0x80001000 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0x???800?? & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0x80000000 & 0x80000000 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

According to Maitra et al. [10] given the value for  $\Delta_{14}^{(0)}$  is 0x???80040, but we found mistake and the value for  $\Delta_{14}^{(0)}$  is 0x???800??. The calculation of  $\Delta_{14}^{(0)}$  is as follows.

$$\Delta_{14}^{(0)} = [s_{14}^{(1)} \oplus ((s_{10}^{(0)} + s_6^{(0)}) <<< 7)] \oplus [s_{14}^{(1)} \oplus (s_{10}^{(0)} + (s_6^{(0)} \oplus 0x80001000)) <<< 7]$$

Here  $s_6^{(0)} \oplus 0x80001000$  have differential in MSB and 12<sup>th</sup> bit position. But due to carry propagation to the left of the bit position 12,  $(s_{10}^{(0)} + s_6^{(0)})$  become 0x????1000. After 7 bit left rotation, the differential in 12<sup>th</sup> bit position moves to 19<sup>th</sup> bit position, which is 0x???800??. The value of '?' depends on  $s_6^{(0)}$ . So  $\Delta_{14}^{(0)}$  should be 0x???800?? instead of 0x???80040.

## 4 Our Work

### 4.1 Characterization of Valid States

We have revisited the Maitra et al. [9] considering two bits differences in two words in first round, from this they obtain valid initial state by reversing one round.

We found mistake in Maitra et al. [9] work while characterizing the valid initial state as explained above (Sect. 3).

Maitra et al. [9], as mentioned that it would be an interesting combinatorial problem to characterize all such states. Based upon that we have characterized nine more values which lead to valid initial states. The combinations  $(s_4, s_7)$ ,  $(s_2, s_3)$ ,  $(s_{13}, s_{14})$ ,  $(s_1, s_6)$ ,  $(s_1, s_{11})$ ,  $(s_1, s_{12})$ ,  $(s_6, s_{11})$ ,  $(s_6, s_{12})$  and  $(s_{11}, s_{12})$  which characterized as valid states. Detail calculation procedure for  $(s_4, s_7)$  is as follows.

$$1. (s_4, s_7) = (0x80000000, 0x80000000)$$

Differential after first round and Transpose:

$$\Delta^{(1)} = \begin{bmatrix} 0 & 0 & 0 \\ 0x80000000 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} (\Delta^{(1)})^T = \begin{bmatrix} 0 & 0x80000000 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0x80000000 & 0 & 0 \end{bmatrix}$$

Differential in reverseround( $s_1, s_5, s_9, s_{13}$ ) works as follows.

$$\Delta_5^{(0)} = [s_5^{(1)} \oplus ((s_1^{(1)} + s_{13}^{(1)}) \ll 18)] \oplus [s_5^{(1)} \oplus ((s_1^{(1)} \oplus 0x80000000) + (s_{13}^{(1)} \oplus 0x80000000)) \ll 18]$$

Here  $s_1^{(1)} \oplus 0x80000000$  and  $s_{13}^{(1)} \oplus 0x80000000$  have differential in MSB. Now adding  $s_1^{(1)} \oplus 0x80000000$  and  $s_{13}^{(1)} \oplus 0x80000000$  and modulo  $2^{32}$  becomes  $s_1^{(1)} + s_{13}^{(1)}$  will result zero. So  $\Delta_5^{(0)}$  is zero. This proves that the state  $\Delta^{(0)}$  is a valid initial state.

$$\Delta_1^{(0)} = [s_1^{(1)} \oplus ((s_{13}^{(1)} + s_9^{(1)}) \ll 13)] \oplus [(s_1^{(1)} \oplus 0x80000000) \oplus ((s_{13}^{(1)} \oplus 0x80000000) + s_9^{(1)}) \ll 13]$$

Here  $s_{13}^{(1)} \oplus 0x80000000$  and  $s_9^{(1)}$  changes in MSB of  $s_{13}^{(1)} + s_9^{(1)}$ . Now  $s_{13}^{(1)} \oplus 0x80000000$  perform left rotation of 13 bits changes the 13<sup>th</sup> least significant bit of  $s_{13}^{(1)} + s_9^{(1)}$  to 0x00001000. Now differential moves to 12<sup>th</sup> bit position. Then XORed with  $s_1^{(1)} \oplus 0x80000000$ , results 0x80001000. So  $\Delta_1^{(0)}$  is 0x80001000.

$$\Delta_{13}^{(0)} = [s_{13}^{(1)} \oplus ((s_9^{(1)} + s_5^{(0)}) \ll 9)] \oplus [(s_{13}^{(1)} \oplus 0x80000000) \oplus (s_9^{(1)} + s_5^{(0)}) \ll 9]$$

By performing addition for  $s_9^{(1)} + s_5^{(0)}$  will result zero. Then XORed with  $s_{13}^{(1)} \oplus 0x80000000$  results to 0x80000000. So  $\Delta_{13}^{(0)}$  is 0x80000000.

$$\Delta_9^{(0)} = [s_9^{(1)} \oplus ((s_5^{(0)} + s_1^{(0)}) \ll 7)] \oplus [s_9^{(1)} \oplus (s_5^{(0)} + (s_1^{(0)} \oplus 0x80001000)) \ll 7]$$

Here  $s_1^{(0)} \oplus 0x80001000$  have differential in MSB and 12<sup>th</sup> bit position. But due to carry propagation to the left of the bit position 12,  $(s_5^{(0)} + s_1^{(0)})$  become 0x????1000. After 7 bit left rotation, the differential in 12<sup>th</sup> bit position moves to 19<sup>th</sup> bit position, which is 0x???800??. The value of '?' depends on  $s_1^{(0)}$ . So  $\Delta_9^{(0)}$  is 0x???800??. Resultant matrix:

$$\Delta^{(0)} = \begin{bmatrix} 0 & 0x80001000 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0x???800?? & 0 & 0 \\ 0 & 0x80000000 & 0 & 0 \end{bmatrix}$$

By following same above procedure we have characterized the other valid states as  $(s_2, s_3), (s_{13}, s_{14}), (s_1, s_6), (s_6, s_{12}), (s_{11}, s_{12}), (s_1, s_{12}), (s_6, s_{11})$  and  $(s_1, s_{11})$  which results valid constants and are as follows.

2.  $(s_2, s_3) = (0x80000000, 0x80000000)$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0x???800?? & 0 & 0 & 0 \\ 0x80000000 & 0 & 0 & 0 \\ 0x80001000 & 0 & 0 & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0 & 0x80000000 & 0x80000000 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

3.  $(s_{13}, s_{14}) = (0x80000000, 0x80000000)$

$$\begin{bmatrix} 0 & 0 & 0 & 0x???800?? \\ 0 & 0 & 0x80000000 \\ 0 & 0 & 0x80001000 \\ 0 & 0 & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0x80000000 & 0x80000000 & 0 \end{bmatrix}$$

4.  $(s_1, s_6) = (0x80000000, 0x80000000)$

$$\begin{bmatrix} 0 & 0x00001000 & 0 & 0 \\ 0x???80000 & 0 & 0 & 0 \\ 0x00000100 & 0x???80000 & 0 & 0 \\ 0x00001000 & 0x00000100 & 0 & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0x80000000 & 0 & 0 \\ 0 & 0 & 0x80000000 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

5.  $(s_1, s_{11}) = (0x80000000, 0x80000000)$

$$\begin{bmatrix} 0 & 0 & 0x00000100 & 0 \\ 0x???80000 & 0 & 0x00001000 & 0 \\ 0x00000100 & 0 & 0 & 0 \\ 0x00001000 & 0 & 0x???80000 & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0x80000000 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0x80000000 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

6.  $(s_1, s_{12}) = (0x80000000, 0x80000000)$

$$\begin{bmatrix} 0 & 0 & 0 & 0x???80000 \\ 0x???80000 & 0 & 0 & 0x00000100 \\ 0x00000100 & 0 & 0 & 0x00001000 \\ 0x00001000 & 0 & 0 & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0 & 0x80000000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0x80000000 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$7. (s_6, s_{11}) = (0x80000000, 0x80000000)$$

$$\begin{bmatrix} 0 & 0x00001000 & 0x00000100 & 0 \\ 0 & 0 & 0x00001000 & 0 \\ 0 & 0x??80000 & 0 & 0 \\ 0 & 0x00000100 & 0x??80000 & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0x80000000 & 0 \\ 0 & 0 & 0 & 0x80000000 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$8. (s_6, s_{12}) = (0x80000000, 0x80000000)$$

$$\begin{bmatrix} 0 & 0x00001000 & 0 & 0x??80000 \\ 0 & 0 & 0 & 0x00000100 \\ 0 & 0x??80000 & 0 & 0x00001000 \\ 0 & 0x00000100 & 0 & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0x80000000 & 0 \\ 0 & 0 & 0 & 0 \\ 0x80000000 & 0 & 0 & 0 \end{bmatrix}$$

$$9. (s_{11}, s_{12}) = (0x80000000, 0x80000000)$$

$$\begin{bmatrix} 0 & 0 & 0x00000100 & 0x??80000 \\ 0 & 0 & 0x00001000 & 0x00000100 \\ 0 & 0 & 0 & 0x00001000 \\ 0 & 0x??80000 & 0 & 0 \end{bmatrix} \Leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0x80000000 \\ 0x80000000 & 0 & 0 & 0 \end{bmatrix}$$

## 4.2 Attack on Salsa and ChaCha

In FSE 2008, Aumasson et al. [1] attack for 128-key bit of Salsa20/7 within  $2^{111}$  time and ChaCha6 in within  $2^{107}$  time. After this with best of our knowledge there does not exist any improvement on this attack. In this paper we have attacked 128-key bit of Salsa20/7 within  $2^{107}$  time and ChaCha6 within  $2^{102}$  time. Details of the calculation is as follows

**Attack on 128-bit Salsa20/7.** In FSE 2008, Aumasson et al. [1] use the differential  $(\Delta^{(4)}_{1,14} \mid \Delta^{(0)}_{7,31})$  with  $|\epsilon_d^*| = 0.130$  and  $\gamma = 0.4$ . With this obtained number of PNBs  $n=38$ ,  $|\epsilon_a^*| = 0.045$  and  $|\epsilon^*| = 0.00592$ . For  $\alpha = 21$ , results in time  $2^{111}$  and data  $2^{21}$ . But the list of the PNBs are no mentioned. By considering same differential and  $|\epsilon_d^*$  we find number of PNBs  $n=35$  with  $|\epsilon_a^*| = 0.040$  and  $|\epsilon^*| = 0.0052$ . For  $\alpha = 21$ , results in time  $2^{114}$  and data  $2^{21}$ . The list of 35 PNB's and the corresponding  $\gamma_k$ 's are shown in Table 1.

In 128-key bit of Salsa20/7 for  $\gamma = 0.3$  results in additional 5 PNBs shown in Table 2. and for  $\gamma = 0.2$  results in additional 11 PNBs shown in Table 3.

In Table 4. we have compared the results with different  $\gamma$  values for the attack on 128-key bit of Salsa20/7.

**Attack on 128-bit ChaCha6.** In FSE 2008, Aumasson et al. [1] use the differential  $(\Delta^{(3)}_{11,0} \mid \Delta^{(0)}_{13,13})$  with  $|\epsilon_d^*| = 0.026$  and threshold  $\gamma = 0.5$ , obtained number of PNBs  $n = 51$ ,  $|\epsilon_a^*| = 0.013$  and  $|\epsilon^*| = 0.00036$ . For  $\alpha = 26$ , results in time  $2^{107}$  and data  $2^{30}$ . But the list of PNBs are not mentioned. By

**Table 1.** Key-bits and the corresponding  $\gamma_k$ 's for the 35 PNBs of the Salsa 7-round attack.

0	1	18	19	20	21	22	23	24	25	26	27	28	29	30
0.59	0.42	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0.98	0.97	0.97
31	33	34	39	43	44	45	57	58	59	65	71	72	73	96
0.96	0.61	0.41	0.45	0.75	0.62	0.45	0.71	0.59	0.45	0.52	0.73	0.60	0.44	0.59
97	110	111	112	116										
0.40	0.76	0.61	0.41	0.41										

**Table 2.** Key-bits and the corresponding  $\gamma_k$ 's for the additional 5 PNBs of the Salsa 7-round attack.

60	66	78
0.30	0.30	0.39
92	124	
0.30	0.30	

**Table 3.** Key-bits and the corresponding  $\gamma_k$ 's for the additional 11 PNBs of the Salsa 7-round attack.

2	6	12	35	40	46
0.23	0.26	0.20	0.20	0.23	0.26
53	74	79	113	117	
0.25	0.26	0.21	0.20	0.24	

considering same requirements we find the time  $2^{106}$  and data  $2^{28}$ . The list of 51 PNB's and the corresponding  $\gamma_k$ 's are shown in Table 5.

In 128-key bit of ChaCha6 for  $\gamma = 0.4$  results in additional 6 PNBs shown in Table 6. and for  $\gamma = 0.3$  results in additional 2 PNBs shown in Table 7.

In Table 8. we have compared the results with different  $\gamma$  values for the attack on 128-key bit of ChaCha6.

### 4.3 Choosing Proper IVs on Salsa and ChaCha

Maitra [8] improved the attack on Salsa20/8 and ChaCha7 by choosing proper IVs corresponding to the 256-key bit. Further improved key search complexity on Salsa20/7 and ChaCha6. Applying the same concept we have attacked 128-key bit of Salsa20/7 within time  $2^{104}$  and ChaCha7 within time  $2^{101}$ . Details of the calculation is as follows

**Attack on 128-bit Salsa20/7.** Considering differential  $(\Delta^{(4)}_{1,14} \mid \Delta^{(0)}_{7,31})$  with  $s_3 = s_{11} = 0$  and  $s_7 = 0xaaaaaaaaaa$ , that is  $k_2 = k_4 = 0$  and

**Table 4.** Different parameters for our attack on 128-key bit Salsa20/7

$\gamma$	n	$ \epsilon_a^* $	$ \epsilon^* $	$\alpha$	Time	Data
0.3	40	0.023	0.0030	21	$2^{110}$	$2^{22}$
0.2	51	0.011	0.0014	21	$2^{107}$	$2^{24}$

**Table 5.** Key-bits and the corresponding  $\gamma_k$ 's for the 51 PNBs of the ChaCha 6-round attack.

2	3	8	9	10	11	12	13	14	15	16	19	20	21	22
0.69	0.50	0.99	0.99	0.99	0.98	0.96	0.93	0.87	0.76	0.56	0.96	0.93	0.88	0.76
23	26	27	28	29	30	31	47	63	72	73	95	96	97	98
0.56	0.94	0.90	0.85	0.76	0.65	0.56	0.59	0.76	0.73	0.53	0.87	0.96	0.93	0.87
99	100	103	104	105	108	109	110	111	112	113	114	115	120	121
0.75	0.55	0.87	0.76	0.56	0.99	0.98	0.97	0.96	0.93	0.87	0.79	0.66	1.0	1.0
122	123	124	125	126	127									
1.0	1.0	1.0	1.0	1.0	1.0									

**Table 6.** Key-bits and the corresponding  $\gamma_k$ 's for the additional 6 PNBs of the ChaCha 6-round attack.

35	51	59
0.44	0.43	0.45
64	88	116
0.42	0.41	0.47

**Table 7.** Key-bits and the corresponding  $\gamma_k$ 's for the additional 2 PNBs of the ChaCha 6-round attack.

39	48
0.33	0.33

$v_1 = 0xaaaaaaaaaa$ . Then  $|\epsilon_d^*| = 0.13225$ . If threshold is  $\gamma = 0.2$ , we find the number of PNBs  $n = 41$ ,  $|\epsilon_a^*| = 0.230$  and  $|\epsilon^*| = 0.0145$ . For  $\alpha = 26$ , results in time  $2^{104}$  and data  $2^{17}$ . The list of 41 PNB's and the corresponding  $\gamma_k$ 's are shown in Table 9.

**Attack on 128-bit ChaCha6.** Considering differential  $(\Delta^{(3)}_{11,0} \mid \Delta^{(0)}_{13,13})$  with  $s_5 = s_9 = 0$  and  $s_{13} = 0xaaaaaaaaaa$ , that is  $k_1 = k_5 = 0$  and  $v_0 = 0xaaaaaaaaaa$ . Then  $|\epsilon_d^*| = 0.041586$ . If threshold is  $\gamma = 0.4$ , we find the number of PNBs  $n = 52$ ,  $|\epsilon_a^*| = 0.0110$  and  $|\epsilon^*| = 0.00045$ . For  $\alpha = 26$ , results in time  $2^{101}$  and data  $2^{28}$ . The list of 52 PNB's and the corresponding  $\gamma_k$ 's are shown in Table 10.

**Table 8.** Different parameters for our attack on 128-key bit ChaCha6

$\gamma$	n	$ \epsilon_a^* $	$ \epsilon^* $	$\alpha$	Time	Data
0.4	57	0.011	0.00030	26	$2^{102}$	$2^{29}$
0.3	59	0.009	0.00025	26	$2^{102}$	$2^{30}$

**Table 9.** Key-bits and the corresponding  $\gamma_k$ 's for the 41 PNBs of the Salsa 7-round attack.

0	1	2	6	18	19	20	21	22	23	24	25	26	27	28
0.59	0.42	0.23	0.25	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0.98
29	30	31	33	34	35	39	40	43	44	45	46	53	57	58
0.98	0.97	0.96	0.61	0.41	0.20	0.46	0.23	0.75	0.62	0.45	0.26	0.25	0.71	0.59
59	60	96	97	110	111	112	113	116	117	124				
0.45	0.30	0.60	0.40	0.76	0.61	0.41	0.20	0.414	0.244	0.30				

**Table 10.** Key-bits and the corresponding  $\gamma_k$ 's for the 52 PNBs of the ChaCha 6-round attack.

2	3	8	9	10	11	12	13	14	15	16	19	20	21	22
0.69	0.50	0.99	0.99	0.99	0.98	0.96	0.93	0.87	0.76	0.56	0.96	0.94	0.88	0.76
23	26	27	28	29	30	31	64	72	73	88	95	96	97	98
0.56	0.94	0.90	0.84	0.76	0.66	0.56	0.42	0.73	0.53	0.41	0.87	0.96	0.93	0.87
99	100	103	104	105	108	109	110	111	112	113	114	115	116	120
0.76	0.55	0.87	0.76	0.56	0.99	0.98	0.97	0.96	0.93	0.87	0.79	0.66	0.47	1.0
121	122	123	124	125	126	127								
1.0	1.0	1.0	1.0	1.0	1.0	1.0								

## 5 Conclusion

In this paper, we explained reverting some differences from the first round leads to valid differentials in the initial state. We characterized different valid states by reversing one round of Salsa20. This can be helpful for producing the sharper biases value. Here we have characterized valid initial states by considering 256-key bit Salsa20. For future work, it will be interesting to characterize valid initial states by considering 128-key bit Salsa20.

We have successfully improved the attack on 128-key bit of Salsa7 and ChaCha6. However our work and previous cryptanalysis result on 128 version of Salsa do not make any threat against Salsa20/12 and Salsa20/20. Finally we hope that these cryptanalysis will result in better understanding of what makes these stream cipher secure.

## References

1. Aumasson, J.-P., Fischer, S., Khazaei, S., Meier, W., Rechberger, C.: New features of Latin dances: analysis of Salsa, ChaCha, and Rumba. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 470–488. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-71039-4\\_30](https://doi.org/10.1007/978-3-540-71039-4_30)
2. Bernstein, D.J.: Chacha, a variant of Salsa20. In: Workshop Record of SASC, vol. 8, pp. 3–5 (2008)

3. Bernstein, D.J.: Snuffle 2005: the Salsa20 encryption function (2015)
4. Choudhuri, A.R., Maitra, S.: Significantly improved multi-bit differentials for reduced round Salsa and ChaCha. *IACR Trans. Symmetric Cryptol.* **2016**(2), 261–287 (2017)
5. Crowley, P.: Truncated differential cryptanalysis of five rounds of Salsa20. In: The State of the Art of Stream Ciphers SASC, vol. 2006, pp. 198–202 (2006)
6. Dey, S., Sarkar, S.: Improved analysis for reduced round Salsa and ChaCha. *Discret. Appl. Math.* **227**, 58–69 (2017)
7. Fischer, S., Meier, W., Berbain, C., Biasse, J.-F., Robshaw, M.J.B.: Non-randomness in eSTREAM candidates Salsa20 and TSC-4. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 2–16. Springer, Heidelberg (2006). [https://doi.org/10.1007/11941378\\_2](https://doi.org/10.1007/11941378_2)
8. Maitra, S.: Chosen IV cryptanalysis on reduced round ChaCha and Salsa. *Discret. Appl. Math.* **208**, 88–97 (2016)
9. Maitra, S., Paul, G., Meier, W.: Salsa20 cryptanalysis: new moves and revisiting old styles. In: 9th International Workshop on Coding and Cryptography, WCC 2015 (2015)
10. Mouha, N., Preneel, B.: Towards finding optimal differential characteristics for ARX: application to Salsa20. Technical report, Cryptology ePrint Archive, Report 2013/328 (2013)
11. Shi, Z., Zhang, B., Feng, D., Wu, W.: Improved key recovery attacks on reduced-round Salsa20 and ChaCha. In: Kwon, T., Lee, M.-K., Kwon, D. (eds.) ICISC 2012. LNCS, vol. 7839, pp. 337–351. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-37682-5\\_24](https://doi.org/10.1007/978-3-642-37682-5_24)
12. Tsunoo, Y., Saito, T., Kubo, H., Suzuki, T., Nakashima, H.: Differential cryptanalysis of Salsa20/8. In: Workshop Record of SASC, p. 12 (2007)



# CloudShare: Towards a Cost-Efficient and Privacy-Preserving Alliance Cloud Using Permissioned Blockchains

Yandong Li<sup>1</sup>, Liehuang Zhu<sup>1</sup>, Meng Shen<sup>1()</sup>, Feng Gao<sup>1</sup>, Baokun Zheng<sup>1,2</sup>,  
Xiaojiang Du<sup>3</sup>, Sheng Liu<sup>4</sup>, and Shu Yin<sup>4</sup>

<sup>1</sup> Beijing Institute of Technology, Beijing, China

{leeyandong, liehuangz, shenmeng}@bit.edu.cn, gaofengbit@foxmail.com,  
zhengbk168@163.com

<sup>2</sup> China University of Political Science and Law, Beijing, China

<sup>3</sup> Temple University, Philadelphia, USA

dxj@ieee.org

<sup>4</sup> Union Mobile Financial Technology Co., Ltd., Beijing, China

{liusheng, yinshu}@umfintech.com

**Abstract.** Data explosion has raised a scalability challenge to cloud storage management, while spinning disk capacity growth rates will continue to slow down. Major data holders such as cloud storage providers with a heavy reliance on disk as a storage medium will need to orchestrate multiple kinds of storage to better manage their relentless data growth.

In this paper, we first explore the scenario that multiple clouds are driven by interests to make the storage resources efficiently allocated without requiring a trusted third party, and then propose a novel model, called CloudShare, to enable multi-clouds to carry out a transparent encrypted data deduplication among cross-users via blockchain. Our scheme significantly reduces the storage costs of each cloud, and saves the upload bandwidth of users, while ensuring data confidentiality and consistency. We demonstrate via simulations on a realistic datasets that CloudShare achieves both the effectiveness and the efficiency.

**Keywords:** Cloud storage · Blockchain · De-duplication  
Cost-efficient · Privacy-preserving · Sharing economy

## 1 Introduction

The advent of cloud storage motivates organizations and ordinary people to outsource data storage to third-party cloud providers. Nevertheless, the fast growth of data volumes in cloud leads to a dramatically increased demand for storage space and upload bandwidth [1]. At the same time, Waldrop [2] predicts the end of *Moore's law* that has powered the information-technology revolution for the

past 50 years, which brings new challenges to cloud storage providers (*CSPs*) to maintain a low cost in a sustainable manner.

The cost of *CSPs* is growing as data grows, whereas cloud storage prices have plummeted over the past few years on account of an ongoing price war among storage service providers [3]. It is estimated that 36% of all data have been stored in the cloud by 2016, compared to just 7% in 2013 due to falling online storage prices [3]. The decrease in barriers to adopt cloud storage further aggravates the storage costs of each *CSP*. Once the provider runs out of capacity, it has to make another sizable investment (in storage media, or network switches and other infrastructure), forming a relatively high marginal cost. Thus, how to reduce the ongoing storage and maintainability cost and re-evaluate the cloud storage strategies will become an urgent thing in the coming future.

Data deduplication is considered as a promising technology to address the challenges of scalability and complexity of enterprise networks and data centers [4]. Network storage service providers may deduplicate by keeping only one or a few copies for each file to reduce unnecessary redundant copies and generating a link to the file for users asking to store the file. It offers secondary cost saving in power and cooling achieved by reducing the number of disk spindles. These savings also translate to lower fees for customers. A constructive concept for remote data storage is cross-user deduplication, in which if two clients upload the same file, the storage server detects the deduplication and stores only a single copy. This kind of deduplication will save both the storage capacity as well as the communication bandwidth. According to a survey by IDC [5], 75% of today's data are duplicated copies, thus deduplication achieves high storage savings. However, sensitive data usually be encrypted for privacy before outsourcing, which makes it at odds to conduct a cross-user deduplication, for the same file may have different encrypted copies in the cloud. Although several effective approaches have been put forward in view of this situation, these solutions are limited to a single cloud storage service. Existing solutions are thoughtless of the fact that popular movies, some applications, backup images, etc., tend to be distributed in different cloud storage, but as far as we know, there are currently no preferable solutions to enable multi-clouds to carry out a transparent encrypted data deduplication among cross-users. Despite the dilemma of this issue, there is also no transparent relation between their benefits and the storage offered by different *CSPs* for them to readily conduct such a co-operation data deduplication.

Inspired by the concept of the *Sharing Economy* [6], we first imagine such a stirring scenario to combine the storage of every cloud to make a *huge alliance cloud* without trusting a center authority, which is supported by the novel technology, called blockchain [7]. The blockchain technology brings us a way that *CSPs* can maintain a tamper-resistant ledger, shared by the participating members, without the need for a trusted third party, potentially making the *CSPs* a possible collaboration in an unprecedented way. In this way, different cloud resources can obtain an effective integration and allocation. Roughly speaking, in our scenario, each cloud serves its own users, but when the client  $C_1$  of the  $CSP_1$  wants to upload a file existing in another cloud, such as  $CSP_2$ . The  $CSP_1$

will learn there is another copy in  $CSP_2$  through blockchain and then only record the ownership of the file instead of storing it. That is, each cloud does not need to store all the files, but presents to have all the files. When the client  $C_1$  downloads the file, it is implicitly download from  $CSP_2$ , and the  $CSP_1$  only need to pay the  $CSP_2$  a very small fee or anything else. For the cloud who store the file, the cloud increased storage resource utilization and can obtain income from other clouds; For the cloud who doesn't store the file, this scheme reduce its storage overhead. For users, due to each file will have a greater probability to appear in the cloud, they do not need to upload the entire file, which saves both bandwidth and uploading time. In this paper, we will try to make a deduplication over encrypted files in such a scenario.

To summarize, we make the following key contributions:

- For the first time, we explore the scenario that *multiple clouds are driven by the interests to work together to achieve an efficient allocation for cloud resource without requiring a trusted third party*, effectively slowing down the gap between data generation speed and storage changes, which is beneficial for the limited storage resources now, and then describe the feasibility of the scene.
- To the best of our knowledge, we first propose a novel model, called *Cloud-Share*, to enable multi-clouds to carry out a transparent encrypted data deduplication among cross-users by adopting a blockchain (ledger) to record the existence and ownership of the file, which is a *privacy-preserving cross-user data deduplication scheme supporting client-side encryption without requiring any additional independent servers*. Our scheme significantly *reduces the storage costs of each cloud, and saves the upload bandwidth of users, while ensuring data confidentiality and consistency*.
- We demonstrate via simulations that CloudShare achieves both the effectiveness and the efficiency.

*Roadmap:* The remainder of this paper is organized as follows. The background and related work are discussed in Sect. 2, followed by the system model, and the threat model discussed in Sect. 3. Section 4 introduces the preliminaries and definitions. We present details of our CloudShare scheme and discuss its security in Sects. 5 and 6, respectively. The design of experiments and results are demonstrated and discussed in Sect. 7. Finally, we present our conclusions and perspectives for future work in Sect. 8.

## 2 Background and Related Work

### 2.1 Blockchain

Blockchain, commonly known as an emerging technology underpinning bitcoin, was first conceptualized by Nakamoto in [8]. It enables an evolving set of parties to maintain a safe, permanent, and tamperproof ledger of transactions without

a central authority, making its potential applications go well beyond enabling digital currencies [9]. Blockchains can be either public, allowing anybody to use them (e.g., bitcoin) or permissioned, creating a closed group of known participants working to provide an immutable ledger that captures the existence of digital facts in a given moment in time in a non-repudiable manner [10]. We can think blockchain a distributed database contains a list of ordered and relevant records called blocks, each comprising a timestamp and a link to a previous block, as well as a set of transactions. Once a participant wants to add a transaction to the ledger, the transaction data will be verified by other participants in the network using cryptographic algorithms. These transactions are broadcast and recorded by each participant in the network, and are finally recorded in a block by a consensus algorithm (e.g. POW [11], PBFT [12]). Once a block is collectively accepted, it is practically impossible to be changed, which make transactions are immutable, trusted, and auditable.

## 2.2 Multi-cloud Storage

Multi-cloud storage is currently an emerging technique using a series of clouds to provide data storage service for clients. Most existing multi-cloud storage systems [13, 14] mainly focus on data reliability regarding cloud failures and vendor lock-ins. MetaStorage [15] and SPANStore [16] provide both integrity and availability guarantees by replicating data across multiple clouds using quorum techniques. However they don't address confidentiality, which is later achieved by Hybris [17] by dispersing encrypted data over multiple public clouds via erasure coding and keeping secret keys in a private cloud. CDstore [18] builds on an augmented secret sharing scheme called convergent dispersal to achieve both bandwidth and storage savings, whereas it does not address consistency issues due to concurrent updates as mentioned in [19].

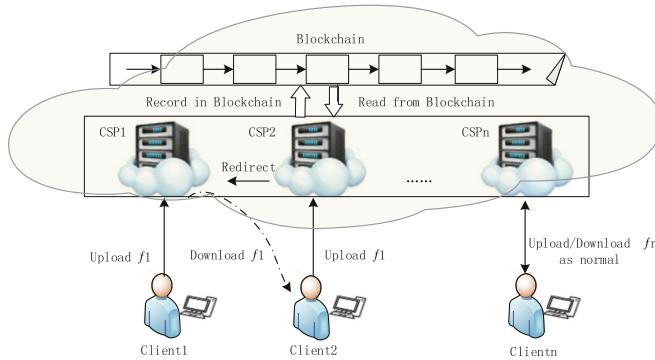
## 2.3 Data Deduplication Security

The demand for data privacy and security is increasing recently [20–26]. Sensitive data usually have to be encrypted before sending to servers, which makes the storage server can neither recognize that the files are identical or coalesce the encrypted files into the space of a single file. Convergent encryption [27] provides confidentiality guarantees for deduplication storage, and has been implemented and experimented in various storage systems. Bellare et al. [28] generalize convergent encryption into Message-locked encryption (MLE) and provide formal definitions of privacy and tag consistency. The same authors also prototype a server-aided MLE system DupLESS [29], which address the applied aspects of encrypted duplication storage. Liu et al. [30] introduce a single-server cross-user deduplication scheme with client-side encryption using a password-authenticated key exchange protocol for MLE key generation. ClearBox [31] enables clients to verify the effective storage space that their data occupies after deduplication. To the best of our knowledge, we are the first to think about the co-operation data deduplication of multiple clouds, and its security.

### 3 A Hierarchical Framework for CloudShare

#### 3.1 System Model

CloudShare is designed for multiple *CSPs* to serve their own group of users, simultaneously, they could readily conduct such a cooperation to save storage cost and obtain some extra earnings from other cloud without reliance on a trusted center party. As shown in Fig. 1, a CloudShare framework can be abstracted as two layers:



**Fig. 1.** A blockchain-based hierarchical architecture for CloudShare

1. the upper layer leverages the blockchain-based trading pattern among *CSPs* (i.e.,  $CSP_1, CSP_2, \dots, CSP_n$ ). Since *CSPs* are the different market entities, we consider each *CSP* interact with such a blockchain, which can be thought as a conceptual party (in reality decentralized) that can be trusted for correctness and availability. Such a blockchain provides a powerful abstraction for the design of distributed protocols. The cloud can write contents into the blockchain and read the contents from it. Once a block is collectively accepted, it is practically impossible to change it or remove it, which is guaranteed by the nature of the blockchain.
2. the bottom layer consists of the *CSPs* and their corresponding users; Users of each *CSP* run the CloudShare client to store their data in their own cloud but may access its data in multiple clouds over the Internet. Each cloud can choose to record the unique identifier of the file (i.e., a cryptographic hash of the content of the file) and its associated information into the blockchain so that other clouds can find the file through it. Once the cloud put the unique identifier of the file into the blockchain. It means it should be responsible for the availability of the files. *CSPs* run the CloudShare server to serve their own users like before. Only one or a few instances of the file is saved and subsequent copies are replaced with a “stub” that points to the original file maybe in another *CSP*. Meanwhile, each cloud may response the requests from the users of another *CSP* to download a file it holds and receive a small fee or anything else for other clearing agreement from that *CSP*.

Taking into account the privacy enforcement for current cloud storage, it is desired to achieve deduplication and encryption simultaneously. The specific approach will be detailed in Sect. 5.

*Remark 1.* Why we choose blockchain to achieve our idea?

In reality, *CSPs* can not fully trust each other in the business environment. The use of blockchain has the following advantages, which are difficult to be satisfied by any other mechanism simultaneously.

- The blockchain protocol is a decentralized protocol framework without requiring a trusted central authority to reach a consensus.
- The blocks can be quickly synchronized across distributed nodes which is very suited for *CSPs* to synchronize a global view of current files.
- The blockchain is practically impossible to be changed once collectively accepted, thus it is undeniable for *CSPs* to tamper with the data, after announcing the possession of the file in the blockchain.

*Remark 2.* What drives *CSPs* to cooperate?

On the whole, once each *CSP* are actively involved in the system, it could use the fixed storage to exchange for more storage space, and the files it is originally supposed to store can help it earn some other incomes. In addition, the encrypted data is more willing to be shared for each *CSP*, which is supported by our scheme. Apparently, cooperation can maximize the profit, so we think the cloud is inclined to cooperate. Of course, some *CSPs* may strongly store more and more additional data, but the storage capacity of each *CSP* is not unlimited, which will eventually achieve a dynamic balance state.

### 3.2 Threat Model

In our CloudShare scheme, we assume the decentralized consensus protocol is secure and the blockchain can be trusted for correctness and availability but not for privacy. We assume data is unpredictable (have high min-entropy) to adversary, not to legitimate clients. We assume the existence of encrypted and authenticated channels(e.g., using SSL/TLS) between the clients and *CSPs*, so as to defend against any eavesdropping activity in the network. We consider the following factors that may impact the security of data stored on cloud servers:

1. An outside adversary plays a role of a client that interacts with *CSPs*, attempting to obtain the ownership of the data without the possession of the whole file.
2. The outside adversary may collude with curious *CSPs* and get access to the storage or the *CSP* itself are curious about the real data of their clients to extract some information about clients' data or the keys about the data while following the protocol correctly.
3. Selfish *CSPs* may potentially misbehave in order to save resources (e.g., deleting or tamper data stored on it, and refused to admit what it had done.), after it announced the possession of the data in the blockchain.

## 4 Preliminaries and Definitions

**Convergent Encryption.** Convergent encryption (CE) is a cryptography scheme that produces identical ciphertext files from identical plaintext files, irrespective of their encryption keys. Thus, it can be used to provide data confidentiality in deduplication. Specifically, a data owner derives a convergent key  $K$  from an original data copy  $M$  and encrypts the data copy with  $K$  to get the ciphertext  $C$ . Beyond that, the data owner also derives a tag  $\tau$  for the data copy, such that  $\tau$  will be used to detect duplicates. Note that the way to produce the tag cannot be used to deduce the convergent key and compromise data confidentiality.

**Definition 1** (*Convergent encryption (CE)*). A convergent encryption scheme can be expressed as the triple  $(\text{KeyGen}, \text{Enc}, \text{Dec})$  such that:

- $\text{KeyGen}(M) \rightarrow K$ , where  $\text{KeyGen}()$  is a cryptographic hash function, taking data  $M$  as inputs and a convergent key  $K$  as output.
- $\text{Enc}(K, M) \rightarrow C$ , where  $\text{Enc}$  is a symmetric encryption algorithm that takes both  $K$  and  $M$  as inputs and then outputs a ciphertext  $C$ .
- $\text{Dec}(K, C) \rightarrow M$ , where  $\text{Dec}()$  is a symmetric decryption algorithm that takes both  $C$  and  $K$  as inputs and then outputs the original data copy  $M$ .
- $\text{Tag}(M) \rightarrow \tau$ , where  $\text{Tag}$  is the tag generation algorithm that takes the original data copy  $M$  or the ciphertext of  $M$  under the encryption algorithm  $\text{Enc}$ , then we get a tag  $\tau$  of the data copy  $M$ .

**Digital Signature.** Digital signature is an identity authentication mechanism which is based on asymmetric cryptography theory. In a signature scheme, a signer first publishes its public key and later signs a message with its private key. Anybody who gets the signed message can utilize the public key of the sender to verify the integrity and nonrepudiation of the message.

**Definition 2** (*Digital Signature*). A signature scheme can be expressed as the triple  $(\text{KeyGen}, \text{Sign}, \text{Vrfy})$  such that:

- $\text{KeyGen}(1^k) \rightarrow (PK, SK)$ , where  $\text{KeyGen}$  is a key generation algorithm taking a security parameter  $k$  as input and outputting a pair of keys  $(PK, SK)$ , which are called the public key and the private key, respectively.
- $\text{Sign}(SK, m) \rightarrow \sigma$ , where  $\text{Sign}$  is a signing algorithm taking a private key  $SK$  and message  $m$  as inputs and outputting a signature  $\sigma$ .
- $\text{Vrfy}(PK, m, \sigma) \rightarrow b$ , where  $\text{Vrfy}$  is a deterministic verification algorithm, taking a public key  $PK$ , a message  $m$ , and a signature  $\sigma$  as inputs and a bit  $b$  as output. When  $b = 1$ , it means the signature  $\sigma$  is valid and  $b = 1$  means invalid.

## 5 The Proposed Scheme: CloudShare

### 5.1 Overview

CloudShare ensures a transparent data deduplication among *CSPs* without compromising the confidentiality of the stored data. We mainly focused on the definition of the two most important operations in cloud storage: storage and retrieval. We expect to combine the novel blockchain technology and convergent encryption techniques. Specially, blockchain enables *CSPs* to synchronize the global view of the files each cloud holds and convergent encryption technique allows cross-users to securely make a data deduplication on ciphertexts. Furthermore, in order to fit multiple cloud applications, the scenario requires users to prove possession of data prior to its upload. The scheme makes full use of the basic cryptographic primitives, making it computationally-efficient to support cross-cloud data deduplication.

### 5.2 Scheme Description

We consider multiple *CSPs* (i.e.,  $CSP_1, CSP_2, \dots, CSP_n$ ) adopt a permissioned blockchain donated by  $B$  as a distributed database.  $CSP_i (i \in [1, n])$  initializes its public/private key pair  $(Pub_i, Priv_i)$ , and publishes the public key  $Pub_i$  to all other *CSPs* in the blockchain network. Also, it initializes a rapid storage system for storing the tags table  $TAB(\text{clients}, \text{tag}, \text{clouds})$  for efficiency. We assume each public key is authenticated by each other. Similar to existing storage providers, CloudShare supports the following operations: Put, Get. In addition, CloudShare supports Proof of Ownership (*POW*), which is used to generate a proof of data possession. For simplify, We will choose two of the clouds to conduct a cross-cloud deduplication as an example to describe the details of CloudShare.

#### *Specification of the Put Procedure*

The Put protocol is executed between the *CSPs* and clients who aim to upload a file  $f$ . Let  $H : \{0, 1\}^* \rightarrow \{0, 1\}^l$  be a cryptographic hash function, where  $l$  represents the token size. Clients initialize a convergent encryption scheme  $(KeyGen, Enc, Dec, Tag)$ , which will be used to encrypt the data of clients and conduct the secure deduplication in the *CSP*. To store a data file in *CSP* (e.g.,  $CSP_1$ ), a client  $\mathcal{C}_1$  and his  $CSP_1$  performs the following operations:

1. For a file  $f$  to be uploaded,  $\mathcal{C}_1$  first generates a hash key  $K_f = KeyGen(f)$ , instead of a random key, derived from  $f$ , where  $KeyGen$  is a (optionally salted) hash function  $H$  (e.g., SHA-256):

$$K_f = H(f) \quad (1)$$

2. To achieve confidentiality,  $\mathcal{C}_1$  encrypts the data file  $f$  with key  $K_f$  to  $f^* = Enc(f, K_f)$ , where  $Enc$  denotes a symmetric key encryption function (e.g., AES-256).  $\mathcal{C}_1$  then saves  $K_f$  in the local place and computes a digital fingerprint  $\tau = H(f^*)$  of  $f$  and save it in the local place.

3. Prior to uploading the file,  $\mathcal{C}_1$  sends  $\tau$  to its  $CSP_1$ , which will serve as a practically unique handle to  $f$ . Here, we take this process as a *POW* scheme. It is initialized for the client to prove its knowledge of the file.
4. Upon receiving  $\tau$ ,  $CSP_1$  will compute  $\tau^* = H(\tau)$ , and check whether there is a  $\tau^*$  in the  $B$ . This process can be divided into two cases according to the query result in  $B$ .
  - (a) *Case1*: Initial Upload of File  $f$ .  
If  $\tau^*$  does not appear in the  $B$ ,  $CSP_1$  responds the result to  $\mathcal{C}_1$  and asks  $\mathcal{C}_1$  to upload the file  $f^*$ .  $CSP_1$  then computes  $\tau^* = H(H(f^*))$ , and record  $\langle \tau^*, PK_1, Sign(SK_1, \tau^*) \rangle$  into the  $B$  and record  $\mathcal{C}_1$  the owner of file  $f$  tagged by  $\tau^*$  as a tuple  $\langle \mathcal{C}_1, \tau^*, PK_1 \rangle$ .
  - (b) *Case2*: Subsequent upload of file  $f$ .  
If the digital fingerprint  $\tau^*$  already exists, the  $CSP_1$  will construct and maintain a set  $TAB$  which contains tuples  $\langle U_F, \tau^*, PK_i \rangle$ , where  $PK_i$  will be extracted from  $B$  as the identity of the  $CSP_i$  corresponding to the file referenced by  $\tau^*$ ,  $U_F$  is the set of clients that are registered to the file.  $\mathcal{C}_i$  will be inserted into  $U_F$ . This tuples means  $\mathcal{C}_i \in U_F$  has the ownership of the file tagged  $\tau^*$  existing in  $CSP_i$ , which is the cloud storage provider that actually holds the file. Note that this can saves storage space at the server and the bandwidth at both sides.

### *Specification of the Get Procedure*

The specification of the Get Procedure is shown as follows. Specifically, when a client  $\mathcal{C}_2$  issues a request to  $CSP_2$  to download a file  $f$  referenced by  $\tau^*$ . It initiates this protocol with its  $CSP_2$ :

1. The  $CSP_2$  first query the tuples referenced by  $\tau^*$  and checks if  $\mathcal{C}_2$  has the ownership of file  $f$ . If this verification passes, it further verifies where the file is. This process can be divided into two cases according to the query result.
  - (a) *Case1*: The file  $f$  is stored in the  $CSP_2$  itself.  
If the  $CSP_2$  itself holds the file, it transforms the file to a package denoted by  $(f^*, \tau^*)$ , where  $f^*$  and  $\tau^*$  are the encrypted contents of  $f$  and its hash tag, respectively. Then, the client can download the encrypted file  $f^*$ .
  - (b) *Case2*: The file  $f$  is stored in  $CSP_i (i = 2)$ .  
If  $CSP_i$  (i.e.,  $CSP_1$ ) holds the file, actions are taken after  $CSP_2$  pays to  $CSP_1$  a very small fee or anything else, such that eventually the client can download the encrypted file  $f^*$  denoted by  $(f^*, \tau^*)$  from  $CSP_1$  and decrypt it to  $f$  with  $K_f$ .
2. Upon receiving the package  $(f^*, \tau^*)$ ,  $\mathcal{C}_2$  fist compute if  $H(H(f^*)) = \tau^*$ . If not, request to download the file again. If the equation is established, then  $\mathcal{C}_2$  can decrypt it to  $f$  with  $K_f$  by computing  $f = Dec(K_f, f^*)$ .

## 6 Security Analysis

We now focus on potential attack scenarios and possible issues that might arise as stated in the threat model Sect. 3.2. Below, we provide an informal security analysis through the following three aspects.

Firstly, an outside adversary without the whole file can play a role of a client that interacts with *CSPs* and it want to obtain the ownership of the data. Notice that if the adversary attempts to be the owner of the file  $f$  without the whole file, it need to prove its knowledge of the file  $f$  by providing  $\tau = H(f^*)$  as stated in scheme description of CloudShare in Sect. 5. Since  $H()$  is a cryptographic one-way hash function, the adversary can not compute a  $\tau = \tau$  when it does not know the  $f^*$  or the  $f$ . Thus, it could not obtain the ownership of  $f$  and download it as well.

Secondly, the adversary may collude with semi-trusted *CSPs* and get access to the storage or the *CSP* itself are curious about the contents of the data belonging to their clients. However, the files have been encoded by the convergent encryption in our scheme before being outsourced to the *CSPs*. Thus, encrypted files cannot be reverted if the adversary could not get the secret keys in convergent encryption. According to the security definition and analysis for the confidentiality in [28], no efficient adversary  $\mathcal{A}$  has non-negligible advantage to distinguish encryption  $f^*$  of unpredictable message  $f$  from random strings  $\Upsilon$ . That is,  $\mathcal{A}$  cannot compromise any private access key or private derivation key for unpredictable files  $f$ . Thus, Cloudshare can also achieve the security for data based on a secure convergent encryption scheme if the encryption key is securely kept by the clients.

Finally, the blockchain can be assumed to make transactions secure, trusted, auditable, and immutable. Once a *CSP* put a transaction  $\langle \tau^*, PK, Sign(SK, \tau^*) \rangle$  into blockchain  $B$ , which is collectively accepted by most of the *CSPs*. The transaction will be testified by irreversible evidences as stated in Sect. 2. Since we adopted secure signature scheme, the signature unforgeability is also obvious. If the *CSP* wants to tamper data stored on it, it requires attacking multiple distributed *CSPs* simultaneously. The consensus protocol featured by blockchain takes by design into account all the *CSPs*. Therefore, there cannot be any database operation completed without most members being aware of it, which ensures data consistency of their clients.

## 7 Evaluation

In this section, we are dedicated to present the experimental evaluation of CloudShare on a realistic datasets. In our implementation, we let the security parameter  $k = 256$  and use the OpenSSL library for all the basic cryptographic primitives. All the CloudShare clients are implemented in Java and the experiment is conducted on some desktop *PCs* which is running windows and equipped with Intel Core I7 processor at 2.40 GHz and 4 GB RAM. As for blockchain, we select the fabric [32] as the underlying technology of the blockchain-based settlement system, which is an open source permissioned blockchain technique hosted by the Linux Foundation. Our implementation of *CSPs* interfaces with aliyun servers, which are equipped with Intel Xeon processor at 2.60 GHz and 8 GB RAM. Each *CSP* is mapped to a node of the blockchain. For a baseline comparison, we also implemented a data deduplication scheme with *CE* for the

single cloud and integrated it with aliyun server, in which clients directly interact with the single *CSP* when storing/fetching their files.

## 7.1 Deduplication Efficiency

An important metric to measure CloudShare is the deduplication efficiency. We thus conduct a evaluation of our scheme compared with the ordinary data deduplication in a single *CSP* using convergent encryption. We use four real-word sets of files (390 GB in total) to evaluate the deduplication efficiency. All the files were obtained from four personal computers  $PC_1, PC_2, PC_3, PC_4$  in our lab.

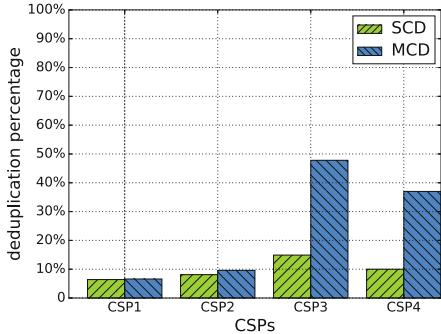
We consider adding a file from a *PC* to its cloud storage as an “upload request”. To generate upload requests, each *PC* will randomly upload its own files from its dataset to its own cloud as far as possible. In our experiment, we map each dataset to a stream of upload requests by generating the requests in random order, where a file that has  $m$  copies generate  $m$  upload requests at random time intervals.  $PC_1, PC_2, PC_3$ , and  $PC_4$  will generate 96335, 87334, 106655, and 77769 upload requests, of which 90123, 80225, 90775, and 69998 are for distinct files, respectively. We define the deduplication percentage  $p$  is:

$$p = \left(1 - \frac{\text{Numbers of all files in storage}}{\text{Total number of upload requests}}\right) * 100\% \quad (2)$$

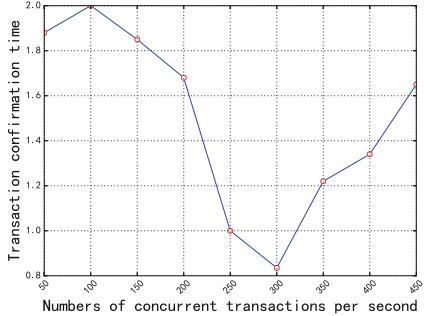
Figure 2 exhibits the deduplication percentage for both single cloud data deduplication (*SCD*) and multiple cloud data deduplication (*MCD*). Note that, it is as expected that *MCD* has a higher deduplication percentage than *SCD* for each *CSP*, due to some daily cooperation and some common media files. In reality, popular files such as movies, applications, backup images tend to be distributed in different cloud storage. That is to say, compared with *SCD*, CloudShare can improve the deduplication efficiency significantly. This also indicates that our method can enjoys lower communication overhead, for users don’t need to upload files existing in other *CSPs*, which becomes more significant when the document set is getting larger.

## 7.2 Performance Evaluation

Blockchain technology can simplify the settlement process, but it may face a performance bottleneck due to the implementation mechanism. In our evaluation, we use fabric to implement the permissioned blockchain and adopt PBFT as the consensus protocol among four *CSPs*, so that the transaction processing speed and transaction confirmation time can be improved significantly. Note that the deduplication process is related to the multiple clouds and multiple clients, in which *CSPs* need to synchronized a global view of current files through the blockchain. As data deduplication mainly involves the interaction with blockchain, we choose not to do the actual upload and download but just test its conformation time to respond the concurrent number of operation requests from various clients, and then take targeted modifications to adapt to the actual needs of the CloudShare.



**Fig. 2.** The deduplication percentage for both *SCD* and *MCD*



**Fig. 3.** The relationship between conformation time and the numbers of concurrent transactions per second

Figure 3 depicts the relationship between conformation time and the numbers of concurrent transactions per second. We run the test 30 times and record the mean of 30 time-consuming value as the result. Note that when the volume of concurrent transactions is relatively small, the system has to wait for the predefined batch time to pack a block. In the second stage (100–300), the transaction confirmation time decreases with the increase of concurrent transactions. This is because the number of transactions is reaching the threshold of quantity to pack a block in fabric. When the transaction volume exceeds the processing capability, there will be some transactions cannot be confirmed in a timely manner which causes that the transaction confirmation time begins to grow.

It can be seen that our implementation was able to achieve 300 concurrent transactions per second and the transaction confirmation time can be maintained within 2 s on the condition that the number of concurrent transactions is less than the maximum capacity, which satisfies the general requirements. In addition, it is worth noting that when the configuration parameters in fabric are fine tuned, the transaction confirmation time can be reduced to less than 1 s to adapt to the actual needs of the CloudShare.

## 8 Conclusion and Future Work

We present, CloudShare, a novel scheme via incorporating the blockchain concept into multi-clouds data deduplication as a promising research topic. In CloudShare, a *CSP* is able to carry out encrypted data deduplication with others in the alliance cooperatively without trusting any trusted third party. More specially, CloudShare greatly saves the cost of storage for multi-clouds and bandwidth for their users, while ensuring data confidentiality and consistency. Extensive security analysis and simulations demonstrate that our proposed scheme satisfies the desired security requirements and guarantees efficiency as well. As part of future work, we plan to investigate how to model the CloudShare via a cooper-

ative game approach to provide an optimization scheme to handle the tradeoff between the storage costs and benefits of each *CSP*.

**Acknowledgments.** This work was supported in part by the National Science Foundation of China under Grant 61602039, in part by the Beijing Natural Science Foundation under Grant 4164098, and in part by the BIT-UMF research and development fund.

## References

1. Sharma, S.: Expanded cloud plumes hiding big data ecosystem. *Future Gener. Comput. Syst.* **59**, 63–92 (2016)
2. Waldrop, M.M.: More than moore. *Nature* **530**(7589), 144 (2016)
3. Paul Rubens. <http://www.enterprisestorageforum.com/storage-management/can-cloud-storage-costs-fall-to-zero-1.html>
4. Akhila, K., Ganesh, A., Sunitha, C.: A study on deduplication techniques over encrypted data. *Procedia Comput. Sci.* **87**, 38–43 (2016)
5. International Data Corporation (IDC). <https://www.emc.com/collateral/analyst-reports/idc-digital-universe-are-you-ready.pdf>
6. Hamari, J., Sjöklint, M., Ukkonen, A.: The sharing economy: why people participate in collaborative consumption. *J. Assoc. Inf. Sci. Technol.* **67**(9), 2047–2059 (2016)
7. Zheng, Z., Xie, S., Dai, H.N., et al.: Blockchain Challenges and Opportunities: A Survey. *Work Pap* (2016)
8. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008)
9. UK Government Chief Scientific Adviser: Distributed ledger technology: beyond block chain. Technical report, UK Government Office of Science (2016)
10. Hardjono, T., Pentland, A.S.: Verifiable Anonymous Identities and Access Control in Permissioned Blockchains (2016). <http://connection.mit.edu/wp-content/uploads/sites/29/2014/12/ChainAnchor-Identities-04172016.pdf>
11. Gervais, A., Karame, G.O., Wüst, K., et al.: On the security and performance of proof of work blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 3–16. ACM (2016)
12. Castro, M., Liskov, B.: Practical Byzantine fault tolerance. In: OSDI, pp. 173–186 (1999)
13. Kotla, R., Alvisi, L., Dahlin, M.: SafeStore: a durable and practical storage system. In: USENIX Annual Technical Conference, Santa Clara, pp. 129–142 (2007)
14. Hu, Y., Chen, H.C., Lee, P.P., Tang, Y.: NCCloud: applying network coding for the storage repair in a cloud-of-clouds. In: Proceedings of the 10th USENIX Conference on File and Storage Technologies, San Jose, p. 21 (2012)
15. Bermbach, D., Klems, M., Tai, S., et al.: MetaStorage: a federated cloud storage system to manage consistency-latency tradeoffs. In: IEEE International Conference on Cloud Computing, pp. 452–459 (2011)
16. Wu, Z., Butkiewicz, M., Perkins, D., Katz-Bassett, E., Madhyastha, H.V.: SPANStore: cost-effective geo-replicated storage spanning multiple cloud services. In: Proceedings of the 24th ACM Symposium on Operating Systems Principles (SOSP 2013), Farmington, pp. 292–308. ACM (2013)
17. Dobre, D., Viotti, P., Vukolić, M.: Hybris: robust hybrid cloud storage. In: Proceedings of the 2014 ACM Symposium on Cloud Computing (SoCC 2014), Seattle, pp. 1–14 (2014)

18. Li, M., Qin, C., Li, J., Lee, P.P.: CDStore: toward reliable, secure, and cost-efficient cloud storage via convergent dispersal. *IEEE Internet Comput.* **20**(3), 45–53 (2016)
19. Bessani, A., Correia, M., Quaresma, B., André, F., Sousa, P.: DepSky: dependable and secure storage in a cloud-of-clouds. *ACM Trans. Storage (TOS)* **9**(4), 12 (2013)
20. Yao, X., Han, X., Du, X., Zhou, X.: A lightweight multicast authentication mechanism for small scale IoT applications. *IEEE Sens. J.* **13**(10), 3693–3701 (2013)
21. Du, X., Xiao, Y., Guizani, M., Chen, H.H.: An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Netw.* **5**(1), 24–34 (2007)
22. Du, X., Guizani, M., Xiao, Y., Chen, H.H.: A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. *IEEE Trans. Wirel. Commun.* **8**(3), 1223–1229 (2009)
23. Du, X., Guizani, M., Shayman, M.: Implementation and performance analysis of SNMP on a TLS/TCP base. In: Proceedings of the Seventh IFIP/IEEE International Symposium on Integrated Network Management (IM 2001), Seattle, pp. 453–466. IEEE (2001)
24. Du, X., Chen, H.H.: Security in wireless sensor networks. *IEEE Wirel. Commun.* **15**(4), 60–66 (2008)
25. Liang, S., Du, X.: Permission-combination-based scheme for Android mobile malware detection. In: Proceedings of IEEE ICC 2014, Sydney, Australia (2014)
26. Shen, M., Ma, B., Zhu, L., Mijumbi, R., Du, X., Hu, J.: Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection. *IEEE Trans. Inf. Forensics Secur.* **13**(4), 940–953 (2018)
27. Douceur, J.R., Adya, A., Bolosky, W.J., Simon, P., Theimer, M.: Reclaiming space from duplicate files in a serverless distributed file system. In: Proceedings of 22nd International Conference on Distributed Computing Systems, pp. 617–624. IEEE (2002)
28. Bellare, M., Keelvedhi, S., Ristenpart, T.: Message-locked encryption and secure deduplication. In: Johansson, T., Nguyen, P.Q. (eds.) *EUROCRYPT 2013*. LNCS, vol. 7881, pp. 296–312. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-38348-9\\_18](https://doi.org/10.1007/978-3-642-38348-9_18)
29. Bellare, M., Keelvedhi, S., Ristenpart, T.: DupLESS: Server-Aided Encryption for Deduplicated Storage. *IACR Cryptology ePrint Archive* 2013, 429 (2013)
30. Liu, J., Asokan, N., Pinkas, B.: Secure deduplication of encrypted data without additional independent servers. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 874–885 (2015)
31. Armknecht, F., Bohli, J.M., Karame, G.O., Youssef, F.: Transparent data deduplication in the cloud. In: ACM SIGSAC Conference on Computer and Communications Security, pp. 886–900. ACM (2015)
32. The Linux Foundation. <https://github.com/hyperledger/fabric/>



# Probability Risk Identification Based Intrusion Detection System for SCADA Systems

Thomas Marsden, Nour Moustafa, Elena Sitnikova<sup>(✉)</sup>, and Gideon Creech

School of Engineering and Information Technology,  
UNSW Canberra, Canberra, Australia

`thomas.marsden@defence.gov.au, nour.moustafa@unsw.edu.au,`  
`{e.sitnikova,g.creech}@adfa.edu.au`

**Abstract.** As Supervisory Control and Data Acquisition (SCADA) systems control several critical infrastructures, they have connected to the internet. Consequently, SCADA systems face different sophisticated types of cyber adversaries. This paper suggests a Probability Risk Identification based Intrusion Detection System (PRI-IDS) technique based on analysing network traffic of Modbus TCP/IP for identifying replay attacks. It is acknowledged that Modbus TCP is usually vulnerable due to its unauthenticated and unencrypted nature. Our technique is evaluated using a simulation environment by configuring a testbed, which is a custom SCADA network that is cheap, accurate and scalable. The testbed is exploited when testing the IDS by sending individual packets from an attacker located on the same LAN as the Modbus master and slave. The experimental results demonstrated that the proposed technique can effectively and efficiently recognise replay attacks.

**Keywords:** SCADA · Security · Network intrusion detection  
MODBUS TCP · Probability risk identification

## 1 Introduction

Research into the security of SCADA systems has grown in recent years, as the potential damage to critical infrastructure including gas, electricity, water, traffic and railway, and/or loss of life and subsequent risk to state security have been realised [19]. SCADA refers to a system of computers and programmable logic controllers (PLCs) that control and monitor industrial plants, processes and machinery [11, 19]. It enables technicians and engineers to supervise and take control of systems remotely [25]. SCADA is commonly employed in systems that are considered critical infrastructure, essential services, in which society relies on in day to day life. More specifically, critical infrastructure is defined by TISN as “Those physical facilities, supply chains, information technologies and communication networks, which if destroyed, degraded or rendered unavailable for an

extended period, would significantly impact on the social or economic wellbeing of the nation, or affect Australia’s ability to conduct national defence and ensure national security” [7, 27]. Unfortunately, most studies have unveiled that security is an afterthought at best in SCADA systems [19]. Various steps taken to mediate the weaknesses have been suggested and are discussed in Sect. 2. We suggest the implementation of an IDS [23, 24]. Commonplace in traditional information technology networks, IDSs are deployed to alert a systems administrator when malicious activity is detected on their network [10]. However, their adoption in SCADA networks has been limited due to a lack of security best practice [15].

Replay attacks are network based attacks where valid data is repeated at a target to cause malicious effect. They are executed by a MitM or intended source, and due to the unencrypted and unauthenticated norm of Modbus TCP, replay attacks are highly effective in exploiting SCADA as these attacks attempt to disrupt the flow of traffic between source and destination [19]. The unencrypted nature permits a replay attack to modify the contents of a Modbus TCP packet [12]. This would manifest itself as modifying values stored in registers, for example if a traffic authority were to take manual control of a set of traffic lights in an accident, a packet directing red lights to ‘turn on’ could be changed to say ‘turn off’ or never arrive at all [11]. The unauthenticated nature of Modbus TCP means that any user on the same LAN as a Modbus slave is able to access memory values of the PLCs and write values to create similar effects [11].

In this study, we suggest a Probability Risk Identification-based (PRI-IDS) for the Modbus TCP protocol named BusNIDS, with the aim of detecting replay attacks. It will achieve this through characterising data with pre-determined risk values, caching periods of data and generating risk values for those cached periods. Caches with risk levels outside of one standard deviation as a threshold from the mean are flagged as potential replay attacks. All information collected from packets is available as header information from the Application Data Unit (ADU) and Protocol Data Unit (PDU), this means that we can deploy this technique in realtime on an encrypted network [9]. A small scale industrial control network detailed in Sect. 4, used for running attacks and testing the IDS and obtaining results comparing with three peer IDS techniques.

The key contributions in this paper are summarised as follows:

- We develop a Modbus TCP IDS that can conduct packet analysis and detect replay attacks.
- We develop a SCADA testbed which is low-cost and simple to configure for evaluating the proposed IDS.
- We compare the performance of the proposed technique with three existing techniques, showing its superiority.

## 2 Background and Literature Review

This section explains the background and previous studies related to the Modbus TCP protocol, IDS for SCADA systems, establishing testbeds for the purpose of IDS research and provides a contextualised review based on aims of the research.

SCADA systems feature a number of security challenges. Bartman and Carson [5] describe SCADA networks with nine primary threat vectors. They are replay attacks, MitM, brute force, dictionary, eavesdropping, Denial of Service (DoS), war dialling, default credentials and data modification. They establish that implementing IPsec to transport data, AES encryption natively implemented into SCADA protocols and the use of IDSs reporting to a centralised Security Information and Event Management (SIEM) are three of the most effective methods to secure a SCADA network [5]. We discuss that the implementation of an IDS that analyses Modbus TCP header data to detect risks, and as such can improve the security posture of a system by working in tandem with encryption [9].

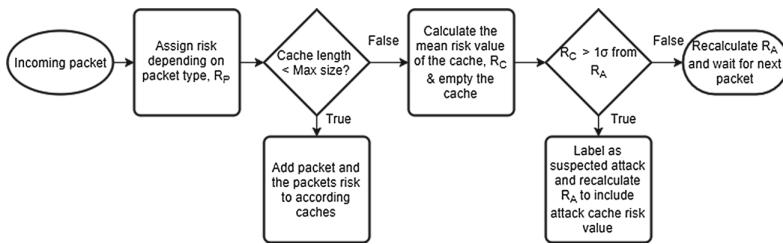
A review of current literature regarding IDS solutions that support Modbus TCP is required to form a baseline knowledge and assists in revealing where novelties for IDS research lay. Yüksel *et al.* [30] develop an anomaly based engine that they test utilising Modbus TCP, Modbus RTU and Siemens S7 datasets. The anomaly based engine is analysing individual TCP packets, determining a probability that the packet is normal traffic based on features of the packet (Source IP, Destination IP, Functions Codes, Read/Write values, etc). Packets with probability deviation beyond a set limit are categorised as threats and generate an alert. The engine is trained faster than existing datasets and is able to detect malicious traffic faster by a factor of 30% over its closest competitor. The trade off for these results is that when using live data rather than a pre-recorded dataset, each packet takes 0.7 ms to parse [30]. Additionally, deploying the technique is limited in scope, as payload data is potentially encrypted and data available to the engine is limited. We demonstrate that through the inclusion of packet analysis and Modbus TCP session analysis, that a Modbus TCP IDS provides an improvement to the detection of malformed packets and replay attacks over existing algorithms trained using machine learning datasets.

Research in the SCADA field has been invested into developing accurate testbed systems at reduced cost [2]. Ahmed *et al.* [1] developed a SCADA system testbed for cybersecurity and forensic research which models a gas pipeline, wastewater treatment plant and power distribution centre. Each plant is modelled using a different brand of in-service PLC, and thus a separate protocol from each plant to a centralised HMI. Due to the testbed utilising currently employed protocols and hardware, vulnerabilities discovered in it are likely to be reproduced in deployed systems [1]. An alternative approach is suggested by Morris *et al.* [2]. The research showed that a completely simulated and virtualised system could compare in accuracy of control and reception of data to a physical system. Genge *et al.* [13] propose a hybrid simulation and emulation framework. PLCs and HMIs are emulated on the Emulab platform, allowing for virtual remapping of physical hardware. The distinct advantage of this method is that it allows for rapid expansion of a system, i.e., 100s of PLCs can be stood up rapidly in the system [13]. The authors showed that a physical test network has the attraction of being cheap, accessible and accurate. However, it does not demonstrate the same scalability as virtualised alternatives.

There are multiple pieces of research dedicated to datasets for the testing of IDSs. Morris and Fernandez [22] have created datasets for a laboratory gas pipeline and water storage tank. Lemay and Fernandez [18] take the view that the current datasets are too limited and are slanted towards the detection of malformed packets. Thus, they devise a set of network data that includes normal operation, operator manipulation, Metasploit exploits, unauthorised remote read/write commands, network scanning and exploitation of their covert Modbus TCP channel for command and control (C2) [18]. They simulated a small electrical network as a plant. The datasets provided by Lemay and Fernandez [18] are publicly available and well rounded. The authors tested the Modbus TCP IDS with individual malicious commands interceding normal network operation.

### 3 Probability Risk Identification (PRI-IDS) Technique

The PRI-IDS technique is designed based on computing a risk level for network data. The following process is summarised as a flowchart in Fig. 1 that demonstrates how our technique works for detecting abnormal events from SCADA data.



**Fig. 1.** Flowchart summarising the PRI-IDS technique

More importantly, we assign a base risk probability,  $R_P$  (Risk value of a packet), to each packet depending on its function code according to Table 1. The risk values assigned to each function code are dependant on the potential effect that the function code has on the system and can be changed depending on the environment that the IDS is deployed in. Each risk value is assigned in real-time as packets are processed. The  $R_P$  value is increased an additional amount if the packet is identified as erroneous. The magnitude of risk values are dependent on the system in which the IDS is configured. As there are few cases in which the traffic light system should see write requests, these packets are more heavily weighted.

A configurable number of packets (CACHE\_MAX\_SIZE) are appended to a cache. The average risk probability value for the cache is calculated as  $R_C$  (Risk value of a cache). Once the cache reaches CACHE\_MAX\_SIZE, the risk value for that cache is stored and used to calculate a moving average,  $R_A$  (Moving average risk). The cache is cleared and progressively filled to its maximum size.

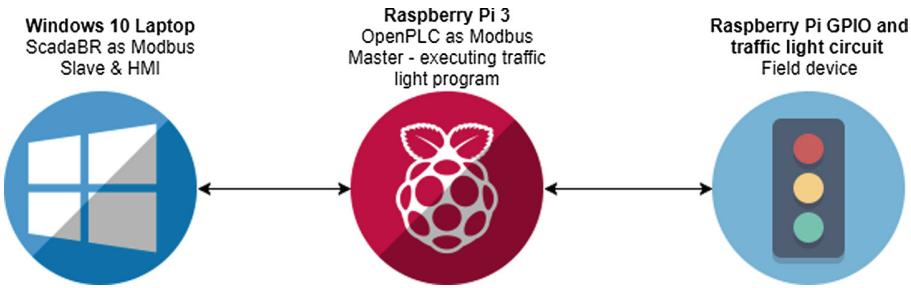
**Table 1.** Risk allocation for Modbus function codes

Function type		Function name	Function code	Risk
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs 2	0.1
		Internal Bits or Physical Coils	Read Coils 1	0.5
			Write Single Coil 5	0.9
			Write Multiple Coils 15	0.9
	16-bit access	Physical Input Registers	Read Input Registers 4	0.1
		Internal Registers or Physical Output Registers	Read Multiple Holding Registers 3	0.5
			Write Single Holding Register 6	0.9
			Write Multiple Holding Registers 16	0.9
			Read/Write Multiple Registers 23	0.9
			Mask Write Register 22	0.5
	File Record Access		Read FIFO Queue 24	0.1
			Read File Record 20	0.1
			Write File Record 21	0.5

At every cache update, the  $R_C$  values are checked. Values which fall greater than  $1\sigma$  (Standard deviation, which is a threshold of identifying malicious events) from the mean  $P_{RC}$  are flagged as potential replay attacks. The corresponding packets which were cause for the deviation are also flagged and made available to the operator. Caches which cause flags to be raised are stored to file in .PCAP formats for further analysis and added as a stored cache. On cache updates, the current cache is also checked against stored caches for matches in packet risk data.

## 4 Testbed and Implementing Proposed PRI-IDS Technique

The testbed developed to run as a Modbus TCP network has three primary components. These components are the Raspberry Pi (RPi) running the openPLC software, traffic light program running on the PLC with a physical traffic light circuit interfacing via the RPi General Purpose Input Output (GPIO) pins, and a Windows 10 laptop running as a SCADA master via ScadaBR (Fig. 2).



**Fig. 2.** Diagram of components in the SCADA testbed. Assembled using icon sets available at [16]

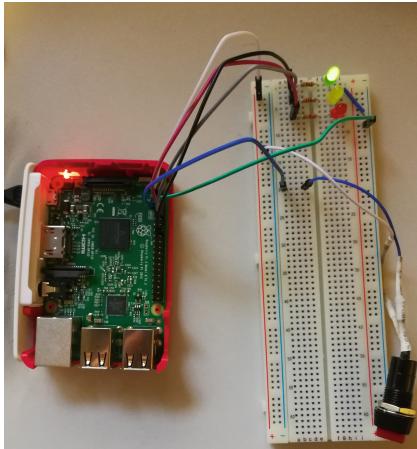
The openPLC software runs on multiple platforms as an effort to build a standardised open source PLC solution [3]. The software can be compiled on a RPi and achieves similar robustness to that of a commercial PLC [21]. At default, openPLC on the RPi operates with a 50 ms scan rate, suitable for gas, water and traffic control SCADA applications [21]. Programs can be uploaded to the PLC through a local web interface. In addition to openPLC, the RPi runs the current iteration of the IDS. The openPLC was programmed with ladder logic to create a simple timer based traffic light program [8].

Fixed memory addresses in the PLC can be accessed via the GPIO pins on the RPi. A traffic light program and accompanying circuit have been constructed as a technique to visualise the effect of replay attacks on the PLC. For example, an attacker can intercept a valid write coils command and replay it at a later time, changing the intended state of the traffic lights [20]. In the physical circuit and HMI, this would present itself as changing the currently active light or disabling them completely. Figure 3 details the physical construction of the field device.

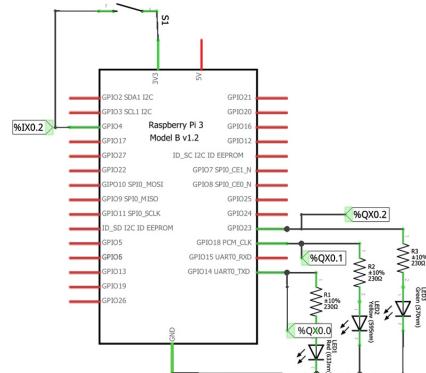
ScadaBR was used as a Modbus master and HMI, accessible via a web interface hosted by the Apache web-server. The laptop used to operate ScadaBR ran Windows 10 with an Intel Core i5-6200U at 2.3 GHz and 8192 MB of DDR4 RAM. ScadaBR is a fully-fledged SCADA master that supports multiple protocols, PLCs, sensors and custom HMIs [29].

The IDS is built in Python 2.7. There are a number of advantages to developing with Python 2.7. The interpreted nature of Python 2.7 enables live testing of code. Commands can be run individually in the interpreter before implementing them into a full script [28]. The Scapy framework is natively built for Python 2.7 and enables full network stack packet reading, writing and sniffing.

The Modbus TCP extension for Scapy enables the author to view packets in terms of Modbus commands rather than hexadecimal data dumps. It is a pivotal tool in the development of the IDS and for testing. It will be used for packet forging to test the IDS [6]. The IDS operates locally on the PLC due to Linux support for Scapy framework being more stable than alternatives. It reports when malformed Modbus TCP packets are sent to the PLC (Fig. 4).



**Fig. 3.** Traffic light circuit connected to RPi PLC via GPIO



**Fig. 4.** Circuit diagram of traffic light and RPi PLC connections

## 5 Experimentation and Discussion

### 5.1 Dataset Used for Experimentation

A simulated attacker is used to test the IDS through a collection of Scapy commands that transmit Modbus TCP packets. These packets are considered the dataset that is used to evaluate the IDS and are sent directly to the PLC. Packets in the dataset are classified as either normal or attack packets. To generate the dataset commands, standard operational traffic between the SCADA master and slave was recorded. The operational traffic featured reading addresses from the PLC, and a write to the PLC to simulate manual override of the system. The traffic was converted to equivalent Scapy commands for repeatability and attacks were added. The malicious commands simulate replay attacks and change the state of coils that are actively used in the system in an unauthorised manner.

### 5.2 Evaluation Metrics and Testbed Description

The IDS is characterised by the metrics accuracy, Detection Rate (DR) and False Positive Rate (FPR). Each of the metrics rely on the state of packets passed through the IDS, these states are defined as the terms true positive (TP), true negative (TN), false positive (FP) and false negative (FN). A packet is classified as TP if it is an attack packet that is correctly identified and thus, TP is the total number of TP classified packets. TN is the total number of packets that the IDS correctly classifies as normal traffic. FP refers to the number of packets that are normal however were classified as attacks by the IDS. FN is the number of attack packets that the IDS incorrectly classifies as normal traffic [23, 24]. The IDS metrics are calculated as functions of these terms according to Eqs. 1, 2 and 3 below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$DetectionRate = \frac{TP}{TP + FN} \quad (2)$$

$$FalsePositiveRate = \frac{FP}{FP + TN} \quad (3)$$

In addition to testing the performance of the IDS, the performance of the Modbus TCP testbed is also considered. The testbed network was operated for a period of 1 h. Every 10 min, write requests to the PLC were made and values in memory were altered. When write requests weren't being made, the PLC was being continuously polled with read requests.

**Table 2.** Errors in Modbus TCP traffic over the testbed for 88125 polls.

Polls	OK	Errors	Error %
88125	88119	5	0.0057

Over the hour, the testbed provided a high level of accuracy. Table 2 displays the errors in Modbus TCP traffic on the testbed over an hour time period.

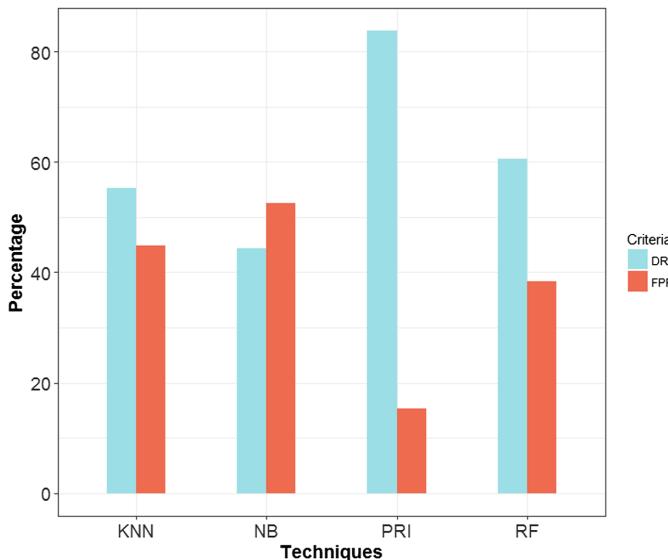
### 5.3 Algorithm Performance Compared with Three Techniques

The performance of proposed algorithm technique was assessed in terms of the overall detection rate and false positive rate, listed in Table 3 and shown in Fig. 5. It is clear that our algorithm achieves the highest DR with roundly 83.7% and the lowest FPR with about 15.3% compared with the techniques [14] of K-Nearest Neighbour, Naïve Bayes and Random Forest. Performance of the alternative techniques used results generated by Hassan *et al.* with technique parameters tuned according to their work [14].

**Table 3.** Comparison of IDS algorithm performances

Technique	DR	FPR
K-Nearest Neighbour (KNN) [14]	55.3%	44.8%
Naïve Bayes (NB) [14]	44.4%	52.6%
Random Forests (RF) [14]	60.5%	38.4%
Probability Risk Identification (PRI)	83.7%	15.3%

The evaluation criteria for machine learning algorithms in IDSs are DR and FPR. Clearly, the results for DR and FPR are improvements over the competing techniques shown in Table 3 and discussed in Hassan's work. The PRI-IDS



**Fig. 5.** Comparing performances of four IDS techniques

produces improved results over the competing algorithms for Modbus TCP due to the customisable risk assignment to individual packets, depending on the potential process of each technique being used for testing. In more details, the K-Nearest Neighbour technique applies a majority vote function between neighbour data points, but as there is relatively a similarity between normal and abnormal data, it cannot achieve better than the PRI-IDS while running in realistic testbed environment [26]. Similarly, the Naïve Bayes and Random Forest techniques cannot find a clear difference between abnormal and normal observation of SCADA data [4, 17].

Ultimately, The PRI-IDS computes the probability of network observations with an accumulating likelihood for prior information, and this finds clear deviations between legitimate and suspicious observations compared with the three techniques. It means that clear distinction can be made between what is desirable and undesirable traffic at the packet level. However, the detection algorithm is susceptible to long, sustained attacks where an attacker transmits packets at a rate that builds the moving average up to a high risk level, allowing high risk packets to be sent without triggering a detection.

## 6 Conclusion

We discussed a Modbus TCP PRI-IDS capable of detecting replay attacks passively at a 28.4% better rate than the next best algorithm, Random Forest, and a corresponding testbed that is cheap, accurate and scalable. Ultimately, we developed an IDS technique designed to detect replay attacks that is more

desirable to implement within a Modbus TCP network than alternatives. The PRI-IDS is written in Python 2.7 and relies on the Scapy network packet manipulation framework. The project is relevant due to the growth in Modbus TCP for communications within SCADA networks. Additionally, the fragility of many SCADA systems means that implementing a passive solution into a network is often preferred over an active or in-line solution. In future work, we will implement the PRI technique into a machine learning environment to test it directly with existing SCADA IDS datasets to determine its performance. It will provide a known and widely used benchmark to test against and will enable fine-tuning of the technique. Once the PRI technique is fine-tuned, it will be reimplemented into a real-time environment to show the increase in performance against the initial iteration of the technique in an IDS.

## References

1. Ahmed, I., Roussev, V., Johnson, W., Senthivel, S., Sudhakaran, S.: A SCADA system testbed for cybersecurity and forensic research and pedagogy. In: ICSS 2016, pp. 1–9. ACM, 6 December 2016
2. Alves, T., Das, R., Morris, T.: Virtualization of industrial control system testbeds for cybersecurity. In: ICSS 2016, pp. 10–14. ACM, 6 December 2016
3. Alves, T.R.: The openPLC project. <http://www.openplcproject.com>
4. Amor, N.B., Benferhat, S., Elouedi, Z.: Naive Bayes vs decision trees in intrusion detection systems. In: Proceedings of the 2004 ACM Symposium on Applied Computing, pp. 420–424. ACM (2004)
5. Bartman, T., Carson, K.: Securing communications for SCADA and critical industrial systems, pp. 1–10. IEEE (2016)
6. Biondi, P.: Welcome to scapy's documentation! 19 April 2010. <http://www.secdev.org/projects/scapy/doc/>
7. Deri, L., SpA, F., Serra, C.: Ntop: a Lightweight Open-Source Network IDS
8. Erickson, K.T.: Programmable logic controllers (2011)
9. Fovino, I.N., Carcano, A., Masera, M., Trombetta, A.: Design and implementation of a secure modbus protocol. Crit. Infrastruct. Protect. **3**, 83–96 (2009)
10. Fovino, I.N., Carcano, A., Masera, M., Trombetta, A.: An experimental investigation of malware attacks on SCADA systems. Int. J. Crit. Infrastruct. Protect. **2**(4), 139–145 (2009)
11. Fovino, I.N., Carcano, A., Murel, T.D.L., Trombetta, A., Masera, M.: Modbus/DNP3 state-based intrusion detection system, pp. 729–736. IEEE Computer Society, Washington, DC (2010)
12. Gao, W., Morris, T., Reaves, B., Richey, D.: On SCADA control system command and response injection and intrusion detection. In: 2010 eCrime Researchers Summit, pp. 1–9, October 2010
13. Genge, B., Nai Fovino, I., Siaterlis, C., Masera, M.: Analyzing cyber-physical attacks on networked industrial control systems. In: Butts, J., Sheno, S. (eds.) ICCIP 2011. IAICT, vol. 367, pp. 167–183. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-24864-1\\_12](https://doi.org/10.1007/978-3-642-24864-1_12)
14. Hassan, M., Moustafa, N., Sitnikova, E., Creech, G.: Privacy preservation intrusion detection technique for SCADA systems. In: Military Communications and Information Systems Conference (MilCIS). IEEE (2017)

15. Iigure, V.M., Laughter, S.A., Williams, R.D.: Security issues in SCADA networks. *Comput. Secur.* **25**(7), 498–506 (2006)
16. JGraph: Draw.io (2017). <https://www.draw.io/>
17. Kim, D.S., Lee, S.M., Park, J.S.: Building lightweight intrusion detection system based on random forest. In: Wang, J., Yi, Z., Zurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3973, pp. 224–230. Springer, Heidelberg (2006). [https://doi.org/10.1007/11760191\\_33](https://doi.org/10.1007/11760191_33)
18. Lemay, A., Fernandez, J.: Providing SCADA network data sets for intrusion detection research. In: USENIX Association, Austin (2016)
19. Mo, Y., Kim, T.H.J., Brancik, K., Dickinson, D., Lee, H., Perrig, A., Sinopoli, B.: Cyber-physical security of a smart grid infrastructure. *Proc. IEEE* **100**(1), 195–209 (2012)
20. Morris, T., Vaughn, R., Dandass, Y.: A retrofit network intrusion detection system for MODBUS RTU and ASCII industrial control systems. In: 2012 45th Hawaii International Conference on System Sciences, pp. 2338–2345, January 2012
21. Morris, T., Alves, T., Das, R.: Virtualization of industrial control system testbeds for cybersecurity, December 2016
22. Morris, T., Thornton, Z., Turnipseed, Z.: Industrial control system simulation and data logging for intrusion detection system research, 19 November 2013
23. Moustafa, N., Creech, G., Slay, J.: Big data analytics for intrusion detection system: statistical decision-making using finite Dirichlet mixture models. In: Palomares Carrascosa, I., Kalutarage, H.K., Huang, Y. (eds.) Data Analytics and Decision Support for Cybersecurity. DA, pp. 127–156. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-59439-2\\_5](https://doi.org/10.1007/978-3-319-59439-2_5)
24. Moustafa, N., Slay, J., Creech, G.: Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. *IEEE Trans. Big Data* (2017)
25. Nicholson, A., Webber, S., Dyer, S., Patel, T., Janicke, H.: Scada security in the light of cyber-warfare. *Comput. Secur.* **31**(4), 418–436 (2012)
26. Peterson, L.E.: K-nearest neighbor. *Scholarpedia* **4**(2), 1883 (2009)
27. Puketza, N.J., Zhang, K., Chung, M., Mukherjee, B., Olsson, R.A.: A methodology for testing intrusion detection systems. *IEEE Trans. Softw. Eng.* **22**(10), 719–729 (1996)
28. Python: Welcome to python.org. <https://www.python.org/about/>
29. ScadaBR: Scadabr home (2017). <http://www.scadabr.com.br/>
30. Yüksel, Ö., den Hartog, J., Etalle, S.: Reading between the fields. In: SAC 2016, pp. 2063–2070. ACM, 4 April 2016



# Anonymizing $k$ -NN Classification on MapReduce

Sibghat Ullah Bazai<sup>(✉)</sup> , Julian Jang-Jaccard , and Ruili Wang

Institute of Natural and Mathematical Sciences, Massey University,  
Auckland, New Zealand  
`{s.bazai, j.jang-jaccard, r.wang}@massey.ac.nz`

**Abstract.** Data analytics scenario such as a classification algorithm plays an important role in data mining to identify a category of a new observation and is often used to drive new knowledge. However, classification algorithm on a big data analytics platform such as MapReduce and Spark, often runs on plain text without an appropriate privacy protection mechanism. This leaves user's data to be vulnerable from unauthorized access and puts the data at a great privacy risk. To address such concern, we propose a new novel  $k$ -NN classifier which can run on an anonymized dataset on MapReduce platform. We describe new Map and Reduce algorithms to produce different anonymized datasets for  $k$ -NN classifier. We also illustrate the details of experiments we performed on the multiple anonymized data sets to understand the effects between the level of privacy protection (data privacy) and the high-value insights (data utility) trade-off before and after data anonymization.

**Keywords:** MapReduce · Data anonymization ·  $K$ -anonymity  
 $k$ -NN classification

## 1 Introduction

In recent years, we witness big data containing a huge amount of personal data such as seen in the data acquired by government administrations, health insurances, social networking sites and IoT devices, to name a few. This exponential growth of big data has demanded a requirement for a system which can provide powerful computation and other related technologies. A big data processing framework using distributed environment, such as MapReduce and Spark, has been widely used to handle such computation to find insights such as correlation between large datasets using Machine Learning algorithms.

In Machine Learning, classification algorithms play an important role in data mining to identify a category of a new observation of data into a set of predefined classes or groups. The k-Nearest Neighbour ( $k$ -NN) [1] is one of such classification methods. In recent years, we witness the adoption of  $k$ -NN algorithm in distributed environments to overcome the computational intensity of having to compare distances of every single training data point.

However, processing  $k$ -NN in MapReduce raises a number of security issues. In MapReduce, Mappers transform the original input key/value pairs into intermediate

key/value pairs after some calculation while reducers aggregate the intermediate values, compute and write them to an output file. These operations at different stage of MapReduce operations are done on plain text which is vulnerable from unauthorized access that puts users' data at a privacy risk. The unauthorized privacy attack can either directly leak sensitive information or indirectly leak information via composite attacks where the adversary can link users' data, illegally obtained at various stage of MapReduce, with public information available via different sources such as Facebook or Twitter.

Providing privacy guarantee during computations of sensitive data can be achieved using privacy preserving techniques such as  $K$ -anonymity [2].  $K$ -anonymity uses *generalization* to hide individual features (also called attribute) or records (also called as tuples) within a crowd or *suppression* to remove highly sensitive records. The size of crowd is typically determined by a privacy parameter  $K$  group size. The use of  $K$ -anonymity in MapReduce platform to provide a certain level of privacy are found in [3–5]. However, these existing studies do not illustrate how to process a privacy preserving technique such as  $K$ -anonymity to be applied in different data analytics scenarios such as classification.

Extending from our earlier study where we illustrated how to apply a  $K$ -anonymity in aggregation scenario [4], this time we illustrate how one can apply a  $K$ -anonymity in a classification scenario that utilizes  $k$ -NN algorithm. We propose a  $k$ -NN classifier which can run on an anonymized data in the MapReduce platform. To the best of our knowledge, our proposed algorithm in this paper is the first attempt to address the classification implementation on anonymized data in the MapReduce platform. Our main contributions are;

- We illustrate the details of the  $k$ -NN classifier algorithms that can run on an anonymized dataset.
- We demonstrate that it is possible to generate different sets of anonymized data using varying degree of privacy parameters (i.e.,  $K$  group size) either applied in the different number of features or the different number of records in the  $K$ -anonymity algorithm.
- We illustrate that different classification results can be obtained based on the different sets of anonymized data sets.
- We provide the impact in the trade-off between the level of privacy protection (data privacy) and the high-value insights (data utility) on classification before and after different anonymized data.

The rest of the paper is organized as follows. In Sect. 2, we provide the necessary background knowledge needed for the paper. In Sect. 3, we describe the related work. In Sect. 4, we describe the details of data anonymization strategies we use and explain the algorithms needed for Map and Reducer operations. In Sect. 5, the experiments and results are discussed. Finally, we conclude our work and discuss the future work planned ahead of us in Sect. 6.

## 2 Background

### 2.1 *k*-Nearest Neighbour

The *k*-nearest neighbour method (*k*-NN) is one of the most widely used classification algorithm in machine learning. Cover and Hart in 1967 formally introduced the original idea of *k*-NN and its properties [1]. *k*-NN works directly on the actual instances of the training data as it does not require building a model to represent the underlying statistics and distributions of the original training data [1]. *k*-NN is based on learning by analogy, that is, by comparing a given test record with training record sets.

Euclidean Distance (*ED*) is often used to measure the distance of two records where the distance indicates the degree of difference (i.e., if *ED* is small the two records are likely to be similar while two records are different if *ED* is big). The distance measure based on *ED* is defined as (1):

$$D(X, Y) = ||X - Y|| = \sum_{i=1}^p (X(i) - Y(i))^2 \quad (1)$$

where  $X(i)$ ,  $Y(i)$  are the  $i$ th dimension attribute values of vector  $X$ ,  $Y$  respectively. In *k*-NN classification, an output can be seen as a class membership as an object is classified by a majority vote of its neighbours. Thus, a class is typically assigned to the object based on the most common classes observed among its neighbours.

There are many different ways to implement *k*-NN algorithms including where the classification should be performed (e.g., [6] proposed the centralized paradigm where the *k*-NN join is performed on a single centralized server) and looking into improving performance overheads (e.g. Parallelization of *k*-NN algorithm [7]). Especially, many existing approaches have been criticized as the computation costs sharply rise when the number of dimensions and the sizes of training sets become large. The use of MapReduce as a processing platform has been regarded as a practical solution to resolve such criticism. The MapReduce framework takes the input data, depending on the size, it automatically splits the input data into smaller manageable chunks. Each smaller chunk is processed by a map task (also often called a mapper interchangeably). The result of a mapper is summarized as key and value pairs. The output (e.g., values) with the same keys are shuffled and reduced by a reducer function.

### 2.2 *K*-Anonymity

*K*-anonymity is one of the first data anonymization techniques with formal mathematical support as a proof. Sweeney [2] introduced *K*-anonymity in 2002 by stating that without ensuring *K* individuals in aggregation single aggregate statistic should not be published. This definition helps every user being able to hide in *K*-1 crowd [9]. In his definition, Quasi-Identifiers (*QID*) are attributes in a dataset which may be linked to a publicly available dataset. The main goal to achieve *K*-anonymity is to replace *QID*s values with more general values so that *QID*s cannot be linked to an individual.

*K*-anonymity is typically achieved by using two techniques called *generalization* and *suppression* with the aim to decrease *QID*s (i.e., there is less obvious identifier to link individual data). Using *generalization*, more granular values are combined

together to create a broader category. This can be achieved both for numerical variables (e.g., *generalization* the monthly salary of \$56,600, \$52,300, and \$73,320 to a single value “above \$50,000”) and for categorical variables (e.g., generalizing the separate degrees of “bachelor”, “masters”, and “PhD” into a single “higher degree”). *Generalization* replaces the original record attributes with less exact but constant values. *QIDs* becomes generalized to a certain point where a few conclusions can be drawn about their relationship with other records. However, the core art of this technique is to understand as what is the optimal level of *generalization* for a given data because repeated *generalization* could decrease the quality of the entire data set. *Suppression* technique involves the removal of records that violates anonymity standards from the data set entirely rather than chaining the value of the records. Also, it is necessary to take a considerable caution because *suppression* can skew the integrity of a dataset when values are eliminated disproportionately to the original distribution of the data. Most often, *suppression* is used in conjunction with the *generalization* to improve the anonymization efficiency. For example, the records that were not within the boundary of  $K$ -anonymity after *generalization* can be automatically suppressed.

### 3 Related Work

Performing  $k$ -NN to provide performance gain has been extensively studied in the literature [7]. Nevertheless, this work only focuses on the centralized and single-thread approach that is not applicable in many modern day applications which requires a large input data for computation. In [8], the authors reported the nearest neighbour classification with *generalization* applied in a large dataset. The main purpose of generalizing exemplars, which merges data into hyper-rectangles, is to improve speed and accuracy but they do not mention how to handle anonymized data. The study in [9] reported a privacy preserving classification techniques. However, the techniques they use focus on neural network as an underlying algorithm then use homomorphic encryption as a data anonymization technique. The direction they took is quite orthogonal to our work. In [10], the authors propose a new nearest neighbour approach using correlation analysis under a MapReduce framework on a Hadoop platform to address the difficult problem of real-time prediction with very large training data sets. However, using their approach, the performance of an algorithm can be seriously affected if the size of the training samples becomes extremely large. For many modern day uses of  $k$ -NN, the computational and the storage issues has become a critical problem [11]. This is because the new applications of  $k$ -NN requires a rather large storage device to contain the whole training set as well as a large computation support in the classification stage.

Airavat [12] proposed a framework for MapReduce by defining mandatory access control (MAC) with differential privacy (DP) on a secure operating system SELinux. Airavat however describes a data anonymization via DP only with a very strict sensitivity pre-defined value which is only applicable to a specific case of applications where the distribution of the input data and types of operations performed on that data is pre-defined.

## 4 Data Anonymization

In this section, we discuss the details of  $K$ -anonymity on  $k$ -NN classifier for MapReduce operations. Our implementation is done based on the privacy-preserving platform we proposed previously [4]. The privacy preserving mechanism receives the user input data and defines a privacy protection mechanism, in our case  $K$ -anonymity. In the algorithm implementation layer, we choose  $k$ -NN as a classifier, transforms the original data into an anonymized equivalent still retaining the content value so that further analytics can be performed on the anonymized data. We measure the classification error on the privacy and utility measurement layer to understand the privacy and utility trade-off between the original data and the anonymized data.

### 4.1 Dataset and Pre-processing

We use the Adult dataset [13] to demonstrate our study. The dataset consists of personal information records extracted from the US census database. We use the dataset for a classification prediction as whether a given person has a potential to earn an annual income over or under \$50,000. The original Adult data set has six continuous and eight categorical features as seen in Table 1.

**Table 1.** Original adult dataset

Age	Workclass**	Fnlwgt	Edu	Edu-num	Marital-status	Occupation	Relationship	Race	Sex	Capital-gain	Capital-loss	Hours-per-week	Native-country	Income
66*	Private	142624	Assoc-acdm	12	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	5556	0	40	Yugoslavia	>50K
55	Private	160631	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	4508	0	8	Yugoslavia	<=50K
53	Private	153064	5th-6th	3	Married-civ-spouse	Exec-managerial	Husband	White	Male	7688	0	10	Yugoslavia	>50K
51	Private	179479	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Female	3325	0	40	Yugoslavia	<=50K
51	Federal-gov	223206	Doctorate	16	Married-civ-spouse	Prof-specialty	Husband	Asian-Pac-Islander	Male	15024	0	40	Vietnam	>50K

\*a light gray represents an example of a tuple

\*\* a dark gray shade represents an example of a feature

The  $k$ -NN algorithm often processes both categorical features and continuous features [1]. To overcome the difficulty of having to process the string data often found in categorical features, many implementations of  $k$ -NN often require the conversion of the categorical features to discrete numerical features. We adopt the conversion from the work of [14], which utilizes unique numerical labels to convert each categorical value into its numerical counterparts. Using this technique, we transform eight categorical features (workclass, edu, marital-status, occupation, relationship, race, sex, native-country) into numerical features. For example, instead of using a country name such as Cambodia, Canada, China a numeric value is used such as 1 to represent the country Cambodia, 2 to as Canada so on. Table 2 represents the Adult dataset after the conversion of the categorical values.

**Table 2.** Adult Dataset after categorical value conversation to numeric

Age	Work-class**	Fnlwgt	Edu	Edu-num	Marital-status	Occupation	Relationship	Race	Sex	Capital-gain	Capital-loss	Hours-per-week	Native-country	Income
66	3	142624	8	12	3	7	1	5	1	5556	0	40	35	>50 K
55	3	160631	12	9	3	7	1	5	1	4508	0	8	35	$\leq 50$ K
53	3	153064	5	3	3	4	1	5	1	7688	0	10	35	>50 K
51	3	179479	12	9	7	4	2	5	2	3325	0	40	35	$\leq 50$ K
51	1	223206	11	16	3	9	1	2	1	15024	0	40	34	>50 K

## 4.2 $k$ -NN Implementation on MapReduce

This section defines the  $k$ -NN algorithm we implemented in MapReduce operations. Our implementation strategies were inspired by the work on [15]. We use the following  $k$ -NN algorithm to get classification errors before data was anonymized. The general processing of data for MapReduce operations follows.

- Reading data: Consider a training dataset  $TR_s$  and a test dataset  $TS_s$ , they are formed by a number of records  $m$ -th (in  $TR_s$ ) and  $t$ -th (in  $TS_s$ ) respectively. Each training sample  $ST_r$  (line) is read and split as a tuple  $(Tp_1, Tp_2, \dots, Tp_D, W_c)$ , where,  $Tp_E$  represent  $E$ -th feature in  $p$ -th tuple, and  $ST_r$  belongs to a class  $W_c$ , for given  $T_p^{W_c}$  and  $D$  diminutions.
- $k$ -NN training: In order to train the  $k$ -NN algorithm, the training dataset  $TR_s$  should contain the value of  $W_c$  while it is unknown for the test dataset  $TS_s$ . For each test sample  $ST_s$  contained in the  $TS_s$  test dataset, the  $k$ -NN model looks for records whose distance proximity is smallest (i.e., indicating the records are similar) in the  $TR_s$  set. To do this, it computes the Euclidean Distances ( $ED$ ) between  $TS_s$  and all  $ST_r$  of  $TR_s$  (i.e., for each sample of test data set with all the sample of train data set). Whereas the  $k$ -nearest neighbours samples ( $NB_1, NB_2, \dots, NB_k$ ) are obtained by ranking the training samples according to the computed distance.
- Alignment with Mapper operations: To apply this in the MapReduce model, we first organize a mapper to compute the classes  $W_c$  from the distance to the  $k$  nearest neighbours for each test and training data.
- Alignment with Reducer operations: The reduce function is responsible for processing the  $ED$  of the  $k$  nearest neighbours from each map and creates a list of  $k$  nearest neighbours by taking those with minimum distance. Reducer shuffles the distances and examines for majority voting, then to assign the  $W_c$  class for  $TS_s$ .  $k$ -NN mapper and  $k$ -NN reducer are described in more detail as follows:

**$k$ -NN Mapper:** In our implementation of  $k$ -NN for MapReduce, we represent our training set as  $TR_s$  and test dataset  $TS_s$ , both with a random number of records store in Hadoop Distributed File System (HDFS) as single file. The first step Mapper accesses the input file from the HDFS and disjoint  $TR_s$  into given number subsets. The training set  $TR_s$  is split into tuples containing the attributes (also known as features) ( $test\_tp_1, test\_tp_2, \dots, test\_tp_D, W_c$ ), where, each  $test\_tp$  represent one feature of adult data set and  $W_c$  represent as an income class (the feature to be classified). Suppose, we have mappers from 1 to  $n$ , for each of the mapper task, it will create  $TR_{sj}$  from  $1 \leq j \leq n$ , which represent the training set sample  $ST_r$ . It should be noted that partitions of given

processes are sequentially executed, for example,  $\text{mapper}_j$  corresponds to  $j$  data chunk. In other words, each mapper will have its corresponding  $\text{TR}_{sj}$  and a class label  $W_c$  for every  $k$  nearest neighbours. Each training record is divided into a subset of  $\text{TR}_s$  in order to compare each subset with it's  $\text{TS}_s$  to find out a distance  $DC$ . The other small subsets are obtained based on  $k$  (degree of neighbours) and number of records in  $\text{TS}_s$ . Distances are stored in the distance matrix  $DC_j$  pairs as " $\langle$ class, distance $\rangle$ " which can be represent as  $\langle W_c, k\text{-distance} \rangle$  with dimension  $k.m$  (i.e.,  $DC_j$  compute all the distances for each tuple of  $\text{TR}_s$  with all element of  $\text{TS}_s$ ). Each Row  $i$  will have  $W_c$  (classifier value) and  $k$ -nearest distance of class. The row  $i$  will repeat till  $t$  for each  $\text{ST}_s$ . After mapper completes its process, it stores the  $\langle$ key, value $\rangle$  pairs as  $\langle (\text{Mapper}_{ID}, W_c), DC_j \rangle$ , where  $\text{Mapper}_{ID}$  is used to identify the mapper in single reducer. The complete pseudo-code for the  $k$ -NN mapper is described in the following Algorithm 1.

---

**Algorithm 1.  $k$ -NN Mapper**


---

**Input:**  $k$  value,  $\text{TS}_s$ .

**Output:** key as mapper identifier  $\text{Mapper}_{ID}$  and class  $W_c$  value as  $DC_j$

```

1: Create  $\text{TR}_{sj}$  with the instances of split  $j$ .
2: for  $i = 0$  to  $i < \text{size}(\text{TS}_s)$  do
3:   Compute  $k$ -NN ( $\text{ST}_s, i, \text{TR}_{sj}, k$ )
4:   for  $m = 0$  to  $m < k$  do
5:      $DC_j(i, m) = \langle W_c(NB_m), \text{ST}_s(NB_m) \rangle > i$ 
6:   end for
7: end for
8: key =  $\text{Mapper}_{ID}, W_c$ 
9: return ( $\langle$ key,  $DC_j$  $\rangle$ )
```

---

**$k$ -NN Reducer:** Reducer is responsible for selecting most relevant neighbours, examines their classes and finds optimal classes for tuples in  $\text{TS}_s$ . The reducer phase can be divided into following four steps:

1. The setup step reads and allocates the distance matrix  $DC_{reduce}$  of the fix size of  $(\text{TS}_s.k\text{-neighbours})$ .  $DC_{reduce}$  value is assigned once a mapper completes  $DC_j$  and sends the data to reducer. Keys from the mapper is separated as  $\text{Mapper}_{ID}$  and  $W_c$ . The size of the distance matrix is initialized with the total number of  $\text{TS}_s$ . Once the setup is done, it moves to the next reduce step.
2. The reduce merge two sorted lists (i.e., one list containing the distances calculated with class and the other list contains the distances calculated with its neighbours). Thus, for every  $\text{TS}_s$ , every distance is compared to its neighbours one at a time starting from the nearest one and sorted according to distances.
3. The third step is cleanup. The cleanup process receives the list of neighbours for all  $\text{TS}_s$  as (class, distance) in the form of  $DC_{reduce}$  for majority voting in order to identify the predicted classes  $W_{cp}$  for  $\text{TS}_s$ . After the cleanup, the key value pair are redefined as  $\text{TS}_s$  classes  $W_{cp}$  and  $\text{TR}_s$  classes  $W_c$ .

4. The comparison of the classifiers between these two classes are done in the classification error step to compute the error rate of  $k$ -NN for each  $k$  values. Following is the pseudocode that we use in this study for the  $k$ -NN reducer.

---

**Algorithm 2.  $k$ -NN Reducer**


---

**Input:** Size of  $TS_s$ ,  $k$  and  $DC_j$ .

**Output:** actual class  $W_c$  and predicated class  $W_{cp}$  as key and  $Error\_rate$  as value

```

1: for  $i = 0$  to  $i < size(TS_s)$  do                                //setup
2:    $cont = 0$ 
3:   for  $m = 0$  to  $k$  do                               //reduce
4:     if  $DC_j(i, cont). ST_s < DC_{reduce}(i, m). ST_s$  then
5:        $DC_{reduce}(i, m) = DC_j(i, cont).$ 
6:        $cont ++$ 
7:     end if
8:   end for
9: end for
10:   $Error\_rate = 0, TP = 0, TN = 0$ 
11: for  $l = 0$  to  $l < size(TS_s)$  do                // CleanUp
12:    $PredClass_l = MajorityVoting(Classes(DC_{reduce}))$ 
13:    $key = W_{cp}, W_c$ 
14:   If ( $W_{cp} == W_c$ ) then
15:      $TP ++$ 
16:   else
17:      $TN ++$ 
18:   end if
19:   $Error\_rate = (TP - TN) / TP)$            // classification error
20: end for
21: return (< $Error\_rate$ >)

```

---

### 4.3 $K$ -Anonymity with $k$ -NN in MapReduce

In this section, we illustrate how we anonymize the original input data using  $k$ -anonymity technique. We run  $k$ -NN classifier on the anonymized data set and get classification error. We compare the classifications errors obtained from a non-anonymized dataset as well as an anonymized dataset to understand whether there has been any impact on classification error. For this task, we extend the mapper operation in the previous section and produce multiple sets of anonymized data sets.

1. The first step is to read  $TS_s$  into tuples containing the attributes (also known as features) ( $test\_tp_1, test\_tp_2, \dots, test\_tp_D, W_c$ ), where, each  $test\_tp$  represent one feature of adult data set and  $W_c$  represent as an income class (feature to be classified). The second step is to anonymize the number of features ( $\alpha$ ), while  $test\_tp_\alpha$  denote the particular feature to be anonymized.
2. Then the third step is to assign  $K$  group size ( $KG$ ) where  $KG$  is the degree of anonymity (i.e., the number of records to hide in a crowd).

3. The forth step is to calculate an average value on each attribute of  $\alpha$  QIDs which to be anonymized. Now  $\alpha$  values are replaced by the average of each feature.
4. In the last step, we replace the average value against each value of KG in continuous features while the average value is used to find more generalized categorical value for the categorical converted numerical features. The input test data now changes from  $TS_s$  to its anonymized counterpart  $TS_{sa}$ .

The pseudo-code for this description is in the following Algorithm 3.

---

**Algorithm 3. ( $K$ -anonymity for  $k$ -NN Mapper)**

---

**INPUT:**  $K$  group size as  $KG$ ,  $QIDs$  attribute as  $\alpha$ ,  $k$ -NN as  $k$  value,  $TRs$  and  $TSs$ .

**OUTPUT:**  $key$  as mapper identifier  $Mapper_{ID}$  and class  $W_c$  value as  $DC_j$ .

Initialize all variables  $Avg=0$ ,  $l=0$

1: Read  $test\_tp_\alpha$  as classifier from  $TSs$

2: **for each**  $i \in K$  **do**

3:      $KG =$  averages of all  $K[l]$ .

4:     **for each** average **do**

5:          $Noise[l] = Avg - KG[l]$ ;

6:          $test\_tp_\alpha[i] = Noise[i]$

7:          $test\_tp_\alpha$  store value on  $TS_{s1}$

8:     **end for each**

9:     **if** ( $\alpha$  is greater than initialized value) **then**

10:         Goto Step 3 for every  $test\_tp_\alpha$  on  $TSs$

11:         **end if**

12: **end for each**

13: Create  $TR_{sj}$  with the instances of split  $j$ .

14: **for**  $i = 0$  **to**  $i < size(TS_{s1})$  **do**

15:     Compute  $k$ -NN ( $ST_s, i, TR_{sj}, k$ )

16:     **for**  $m = 0$  **to**  $m < k$  **do**

17:          $DC_j(i, m) = \langle W_c(NB_m), ST_s(NB_m) \rangle > i$

18:     **end for**

19: **end for**

20:  $key = Mapper_{ID}, W_c$

21: return ( $\langle key, DC_j \rangle$ )

---

In our study, we observe the effect of an anonymisation in two different aspects: tuple-based vs feature-based *generalization*. We first examine the effect of anonymization by its usual tuple-based (i.e., making the number of records same), secondly, we examine the effect of anonymization by its feature-based (i.e., making the number of features across records same). For the former, we analyze 4 different tuple-based degree  $K = \{5, 10, 100, 1000\}$  for example  $K = \{5\}$  indicates that there will be 5 records made same where  $K = \{10\}$  indicates that there will be 10 records made same and so on. *Simple* represent no anonymisation and transformation is applied on data. For the latter, we analyze 5 different feature-based degree  $Ax = \{2, 4, 8, 12, \text{all}\}$  where  $x$  indicates the number of QIDs. From the feature-based generalization, A2 represents  $\alpha = 2$ , i.e., age and workclass are used as QIDs and generalized whereas A4 represents

$\alpha = 4$ , i.e., the four features age, work-class, fnlwgt, and education are used as  $QIDs$ . A8 and A12 represents in the similar fashion. We use the special notation AA to mean all features are used as  $QIDs$  and generalized accordingly. Table 3 represents the snippet of data anonymization on  $K=5$  degree on different numbers of  $QIDs$  being generalized.

**Table 3.** The sample of  $K=5$  tuple with different number of column generalisation

Age	Work-class**	Fnlwgt	Edu	Edu-num	Marital-status	Occupation	Relationship	Race	Sex	Capital-gain	Capital-loss	Hours-per-week	Native-country	Income
<b>5 – anonymity with on 2 Features age and workclass</b>														
55.2	2.6	142624	8	12	3	7	1	5	1	5556	0	40	35	>50 K
55.2	2.6	160631	12	9	3	7	1	5	1	4508	0	8	35	$\leq 50$ K
55.2	2.6	153064	5	3	3	4	1	5	1	7688	0	10	35	>50 K
55.2	2.6	179479	12	9	7	4	2	5	2	3325	0	40	35	$\leq 50$ K
55.2	2.6	223206	11	16	3	9	1	2	1	15024	0	40	34	>50 K
<b>5 – anonymity with on 4 Features age and workclass, fnlwgt and education</b>														
55.2	2.6	171800.8	9.6	12	3	7	1	5	1	5556	0	40	35	>50 K
55.2	2.6	171800.8	9.6	9	3	7	1	5	1	4508	0	8	35	$\leq 50$ K
55.2	2.6	171800.8	9.6	3	3	4	1	5	1	7688	0	10	35	>50 K
55.2	2.6	171800.8	9.6	9	7	4	2	5	2	3325	0	40	35	$\leq 50$ K
55.2	2.6	171800.8	9.6	16	3	9	1	2	1	15024	0	40	34	>50 K
<b>5 – anonymity with on all Features</b>														
55.2	2.6	171800.8	9.6	9.8	3.8	6.2	1.2	4.4	1.2	0	7220.2	27.6	34.8	>50 K
55.2	2.6	171800.8	9.6	9.8	3.8	6.2	1.2	4.4	1.2	0	7220.2	27.6	34.8	$\leq 50$ K
55.2	2.6	171800.8	9.6	9.8	3.8	6.2	1.2	4.4	1.2	0	7220.2	27.6	34.8	>50 K
55.2	2.6	171800.8	9.6	9.8	3.8	6.2	1.2	4.4	1.2	0	7220.2	27.6	34.8	$\leq 50$ K
55.2	2.6	171800.8	9.6	9.8	3.8	6.2	1.2	4.4	1.2	0	7220.2	27.6	34.8	>50 K

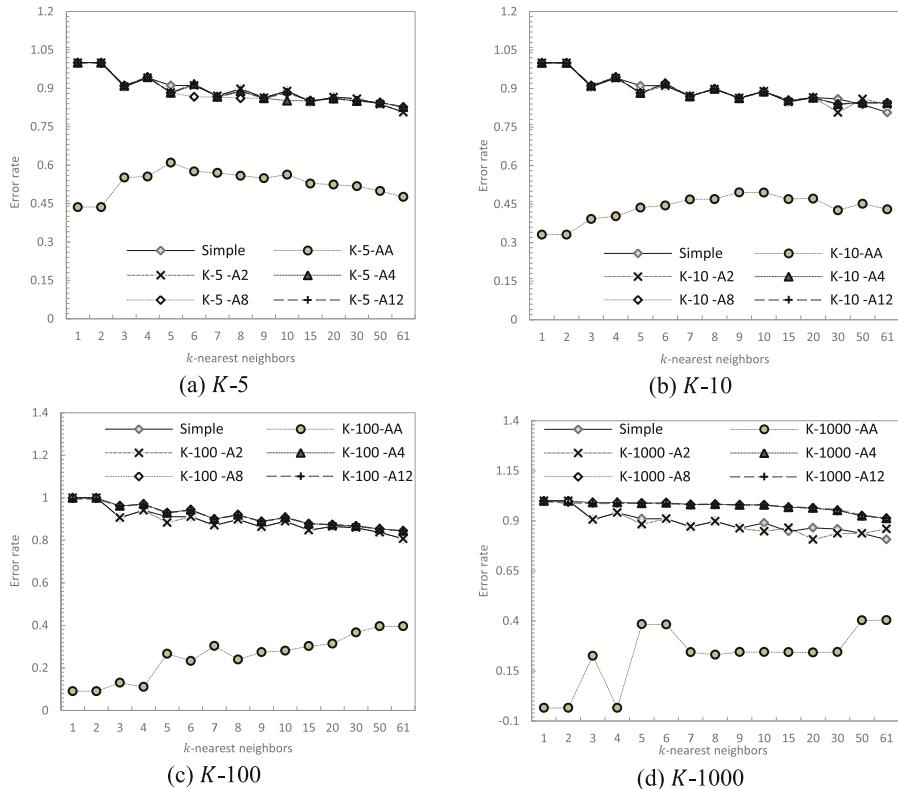
## 5 Experiments and Results

A set of experiments have been conducted on the Adult dataset to observe the  $k$ -NN based classification errors on data anonymized using  $k$ -anonymity. The experiment was performed on the single node cluster with the following specification: (1) the CPU model: Intel(R) Xeon(R) CPU E5-1650 v3, (2) the processing speed: @ 3.50 GHz, (3) the number of core processors: 6, (4) the storage capacity: 4 Tara bytes, and (5) the memory size 32 GB of RAM.

We first run the experiment on both the training and test datasets on  $k$ -NN classifier without any anonymization then check the classification error.

### 5.1 Applying $K$ -Anonymity on $k$ -NN Classifier

Figure 1 illustrates the result on the classification error studying from the feature-based anonymization. For example Fig. 1a shows the results of 5 records anonymized on 2  $QIDs = \{\text{age, workclass}\}$  which is denoted as K-5-A2 while the results of 5 records anonymized on 4  $QIDs = \{\text{age, workclass, fnlwgt, edu}\}$  is denoted as K-5-A4. The same notation is used for other number of  $QIDs$ .



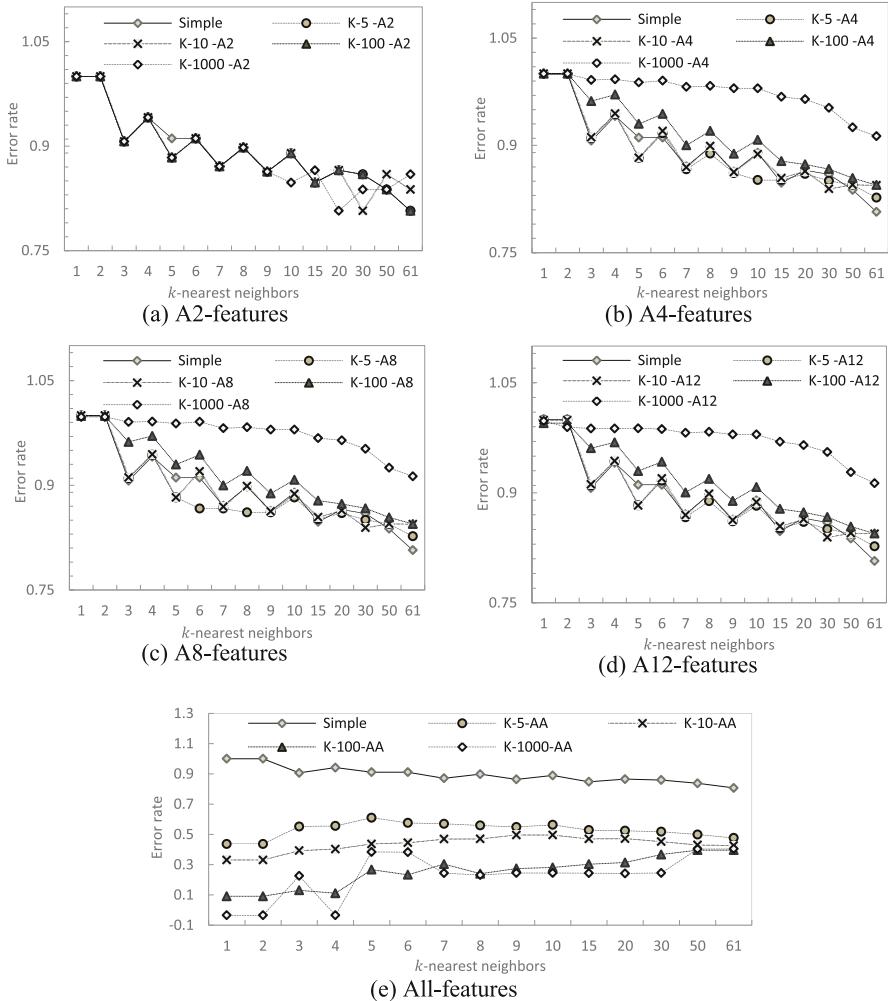
**Fig. 1.** *K*-anonymity on varying degrees of anonymized feature sets

Here is the summary of our observations;

- The number of features being anonymized attributes to the decreasing accuracy (i.e., increasing classification error) as we see this in all graphs.
- As the number of k-nearest neighbours increases, more classification errors are generated. This is due to the increasing size of the sample being the subject of the classification, that is, there is increasing probability of producing an error as there are more data.
- There is a huge amount of classification errors when all features are anonymized in comparison to when there are at least a few features still not anonymized.
- The distribution of the data within a feature affects on the number of classification errors. If the distribution of the data is wide and if they are generalized, they tend to subject to more classification errors.
- With the increasing number of K degrees, the fluctuation of classification errors becomes unpredictable. For example, with K-5 and K-10, we observe a steady increasing or decreasing of classification errors in a smaller range scale which was between 45%–60% with K-5 while 33%–45% with K-10. In the meantime,

classification errors were sharply increased from 10% to 40% in K-100 while it was between 0%–45% with K-1000.

Figure 2 illustrates the result on the classification error studying from the tuple-based–anonymization. For example, the Fig. 2a shows the results of 2 QIDs = {age, work – class} anonymized with different degree of  $K = \{5, 10, 100, 1000\}$ .



**Fig. 2.** QIDs on varying degrees of  $K$ -anonymity

Here is the summary of our observations;

- There is more classification errors produced as the degree of  $K$  increases. It is easy to understand this pattern because simply more data means the increasing possibility

with classification errors. This is observed in all graphs, irrelevant to the number of *QIDs* involved in the anonymization process.

- When only two *QIDs* were anonymized, as shown in Fig. 2a, the effect of increasing  $K$  degree is negligent. As the number of *QIDs* increased to be anonymized, the scale of classification error range becomes wider. For example, in Fig. 2a where it is only two *QIDs* anonymized, there is almost no difference in classification errors among  $K$ -degrees. However, in Fig. 2e where all *QIDs* were anonymized,  $K$ -5 classification errors stay around 50%,  $K$ -10 classification errors stay around 30% whereas  $K$ -100 stays around 10%.
- The utility of anonymized data is higher with a fewer *QIDs* regardless  $K$  degree as we do not see much difference in the classification errors between non-anonymized data and anonymized data.

## 6 Conclusions and Future Work

This research work is an extension from our previous work [4] where we focus running a classification algorithm on the anonymized dataset running a MapReduce platform. In this research we used k-NN as a classifier on anonymised data. We used the measurement of classification errors to observe the effects between privacy verses utility trade-offs when different sets of data were anonymized using multiple privacy parameters of K-anonymity. We used two different approaches; anonymizing the data based on (1) tuples and (2) features. As expected, the number of k-nearest neighbour has the close relationship with classification errors introduced. More data in a dataset produced higher probability of classification errors. We also observed that the distribution of the data within a feature for given dataset affects quite significantly on classification error.

In our future work, we plan to run our experiments in multiple node cluster which may need modification in the algorithms we used in this study. We also plan to make more close observations on the classification errors on different parameters on  $K$ -anonymity and differential privacy such as finding the most optimal point for privacy and utility trade off.

## References

1. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**, 21–27 (1967)
2. Sweeney, L.: K-anonymity: a model for protecting privacy 1. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **10**, 557–570 (2002)
3. Zhang, X., Yang, L.T., Liu, C., Chen, J.: A scalable two-phase top-down specialization approach for data anonymization using mapreduce on cloud. *IEEE Trans. Parallel Distrib. Syst.* **25**, 363–373 (2014)
4. Bazai, S.U., Jang-Jaccard, J., Zhang, X.: A privacy preserving platform for MapReduce. In: Batten, L., Kim, D.S., Zhang, X., Li, G. (eds.) ATIS 2017. CCIS, vol. 719, pp. 88–99. Springer, Singapore (2017). [https://doi.org/10.1007/978-981-10-5421-1\\_8](https://doi.org/10.1007/978-981-10-5421-1_8)

5. Zhang, X., Dou, W., Pei, J., Nepal, S., Yang, C., Liu, C., Chen, J.: Proximity-aware local-recoding anonymization with MapReduce for scalable big data privacy preservation in cloud. *IEEE Trans. Comput.* **64**, 2293–2307 (2015)
6. Stupar, A., Michel, S., Schenkel, R.: RankReduce - processing K-nearest neighbor queries on top of mapreduce. In: CEUR Workshop Proceedings. vol. 630, pp. 13–18 (2010)
7. Zhang, C., Li, F., Jesters, J.: Efficient parallel k NN joins for large data in MapReduce. In: Proceedings of the 15th International Conference on Extending Database Technology - EDBT 2012, p. 38 (2012)
8. Inan, A., Kantarciooglu, M., Bertino, E.: Using anonymized data for classification. In: Proceedings - International Conference on Data Engineering, pp. 429–440 (2009)
9. Barylai, M., Jang-Jaccard, J., Liu, D.: Towards privacy-preserving classification in neural networks. In: 2016 14th Annual Conference on Privacy, Security and Trust, PST 2016, pp. 392–399 (2016)
10. Xia, D., Li, H., Wang, B., Li, Y., Zhang, Z.: A map reduce-based nearest neighbor approach for big-data-driven traffic flow prediction. *IEEE Access*. **4**, 2920–2934 (2016)
11. Zhou, L., Wang, H., Wang, W.: Parallel implementation of classification algorithms based on cloud computing environment. *TELKOMNIKA Indones. J. Elect. Eng.* **10**, 1087–1092 (2012)
12. Roy, I., Setty, S.T.V., Kilzer, A., Shmatikov, V., Witchel, E.: Airavat: security and privacy for MapReduce. In: Proceedings of the 7th USENIX conference on Networked systems design and implementation, p. 20 (2010)
13. Frank, A., Asuncion, A.: UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science, Irvine, CA. 2008, (2010)
14. Inan, A., Kantarciooglu, M., Ghinita, G., Bertino, E.: Private record matching using differential privacy. In: Proceedings of the 13th International Conference on Extending Database Technology - EDBT 2010, p. 123 (2010)
15. Maillo, J., Triguero, I., Herrera, F.: A MapReduce-based k-Nearest neighbor approach for big data classification. In: Proceedings - 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom. vol. 2, pp. 167–172 (2015)



# A Cancellable Ranking Based Hashing Method for Fingerprint Template Protection

Zhe Jin<sup>1</sup>, Jung Yeon Hwang<sup>2</sup>, Soohyung Kim<sup>2</sup>, Sangrae Cho<sup>2</sup>,  
Yen-Lung Lai<sup>1</sup>, and Andrew Beng Jin Teoh<sup>3</sup>

<sup>1</sup> School of IT, Monash University Malaysia, Jalan Lagoon Selatan,  
46150 Bandar Sunway, Selangor Darul Ehsan, Malaysia

{jin.zhe, lai.yenlung}@monash.edu

<sup>2</sup> Electronics and Telecommunications Research Institute (ETRI),  
Seoul, South Korea

{videmot, lifewsky, sangrae}@etri.re.kr

<sup>3</sup> School of Electrical and Electronic Engineering, Yonsei University,  
Seoul, South Korea

bjteoh@yonsei.ac.kr

**Abstract.** Despite a variety of theoretical-sound techniques have been proposed for biometric template protection, there is rarely practical solution that guarantees non-invertibility, cancellability, non-linkability and performance simultaneously. In this paper, a cancellable ranking based hashing is proposed for fingerprint template protection. The proposed method transforms a real-valued feature vector into an index code such that the pairwise-order measure in the hashed codes are closely correlated with rank similarity measure. Such a ranking based hashing offers two major merits: (1) Resilient to noises/perturbations in numeric values; and (2) Highly nonlinear embedding based on the rank correlation statistics. The former takes care of the accuracy performance mitigating numeric noises/perturbations while the latter offers strong non-invertible transformation via nonlinear feature embedding from Euclidean to Rank space that leads to toughness in inversion yet still preserve accuracy performance. The experimental results demonstrate reasonable accuracy performance on benchmark FVC2002 and FVC2004 fingerprint databases. The analyses justify its resilience to inversion, brute force and preimage attack as well as satisfy the revocability and unlink ability criteria of cancellable biometrics.

**Keywords:** Cancellable biometrics · Fingerprint recognition · Rank hashing

## 1 Introduction

Biometrics become commonplace for identity management systems nowadays. The proliferation of biometric systems yields massive number of templates. The security and privacy of biometric template is an escalating concern if compromised. Such a concern is attributed to the strong binding of individuals and privacy, and further complicated by the fact that biometric is irrevocable. Given the above threats, a number of proposals have been reported for protecting the biometric templates. However,

designing a decent biometric template protection (BTP) scheme with the following criteria [1, 2, 7] remain challenge:

- Non-invertibility or Irreversibility: It should be computationally infeasible to derive the original biometric template from a single or multiple protected template and/or the helper data of BTP.
- Revocability or Renewability: A new instance of protected template can be revoked when the existing template is compromised.
- Non-linkability or Unlinkability: It should be computationally difficult to differentiate two or more instances of the protected biometric templates derived from the same biometric trait.
- Performance preservation. The accuracy performance of the protected biometric template should be preserved.

Generally, the BTP schemes in literature can be broadly divided into two categories: cancellable biometrics and biometric cryptosystems. Biometric cryptosystem serves the purpose of either securing the cryptographic key using biometrics (key binding) or directly generating the cryptographic key from the biometrics (key generation) [2]. On the other hand, cancellable biometrics [3] is a more direct solution for BTP as biometric cryptosystem is primary meant to protect secret (such as crypto key) rather than biometric templates. Cancellable biometrics refers to the irreversible transform applied to the biometric template to generate the protected template ensuring the security and privacy of the original biometric template. If a cancellable biometric template is compromised, a new template can be re-generated from the same biometrics.

Several decent review papers exist for BTP such as [2, 4–6]. We focus a few latest and relevant fingerprint related BTP schemes. Ferrara et al. [8] propose a non-invertible scheme for minutia cylinder code (MCC), a state-of-the-art fingerprint descriptor [9], namely protected MCC (P-MCC) via binary principle component analysis. Despite the cancellability is not addressed in P-MCC, a two-factor P-MCC, namely 2P-MCC [10] is later proposed to make P-MCC becomes cancellable. MCC and its successors are dedicated for fingerprint minutiae and thus it is not directly transferred to other popular biometrics such as face and iris.

A generic cancellable biometrics scheme, namely bloom filter has been introduced for iris [11], face [12] and fingerprint [13] recently. Despite the decent performance preservation, the security and privacy of bloom filter based schemes remains open. For instance, Hermans et al. [14] demonstrate a simple and effective attack scheme that matches two protected templates derived from the same IrisCode using different secret bit vectors, thus break the requirement of non-linkability. Bringer et al. [15] further analyzed the non-linkability of the protected templates generated from two different IrisCode of the same subject.

Sandhya and Prasad [16] propose a k-nearest neighbor structure from fingerprint minutia to construct a fixed-length binary vector. The binary vector is Fourier transformed yield a complex vector. Cancellable template can be generated by simply multiplying the complex vector with Gaussian random matrix. This technique yields reasonable recognition accuracy. However, the security of the proposed method is insufficiently analyzed.

Wang and Hu [17] propose a blind system identification approach to protect biometric template. This is motivated by the fact that source signal cannot be recovered if the identifiability is dissatisfied in blind system identification. This new approach exhibits well accuracy performance preservation and the irreversibility of transformed template is justified theoretically and experimentally.

In this paper, we report a new cancellable biometric scheme based on the ranking based hashing, which is inspired from the “Winner Takes All” (WTA) hashing [18] that used for solving the fast similarity search. The proposed scheme enjoys the merits of strong theoretical guarantee of accuracy preservation after hashing. With its pure discrete indices representation nature, a product of non-linearly transformed real-valued biometric features, the scheme can strongly protect the biometric data from being inverted. The analyses justify its resilience to inversion, brute force and pre-image attacks as well as satisfy the revocability and unlinkability criteria of cancellable biometrics. Besides, the implementation is also incredibly simple for practical applications. We demonstrate the feasibility of this method with fingerprint modality.

## 2 Preliminary

The basic idea of WTA hashing [18] is to compute the ordinal embedding of an input data based on the partial order statistics. More specifically, the WTA hashing is a non-linear transformation based on the *implicit order* rather than the absolute/numeric values of the input data, and therefore, offers certain degree of resilience to numerical perturbation while giving a good indication of inherent similarity between the compared items. The overall WTA hashing procedure can be summarized into five steps as follows:

1. Perform  $P$  random permutations on the input vector with dimension  $n$ ,  $\mathbf{x} \in \mathbb{R}^n$ .
2. Select the first  $K$  items of the permuted  $\mathbf{x}$ . Choose the largest element within the  $K$  items.
3. Record the corresponding index values in bits.
4. Step 1–step 3 are repeated  $m$  times, yielding a hash code of length  $m$ , which can be compactly represented using  $m\lceil\log_2 K\rceil$  bits.

WTA is indeed a special instance of Locality Sensitive Hashing (LSH) [24], which is primarily used to reduce the dimensionality of high-dimensional data by hashing the input items so that similar items map to the same “buckets” with high probability where the number of buckets being much smaller than the input items.

Formally, the definition of LSH is given as follows:

**Definition 1.** A LSH is a probability distribution on a family  $H$  of hash functions  $h$  such that  $P[h(X) = h(Y)] = S(X, Y)$ . With a similarity measure function,  $S$  define on the collection of object  $X$  and  $Y$ .

The key ingredient of the LSH is the hashing of object collection  $X$  and  $Y$  by means of multiple ( $m$  to be exact) hash functions  $h_i$ ,  $i = 1, \dots, m$ . The use of  $h_i$  enables approximation of the pair-wise distance of  $X$  and  $Y$  in terms of collision probability. LSH ensures that  $X$  and  $Y$  with high similarity renders higher probability of collision in

the hashed domain; on the contrary, the data points far apart each other result a lower probability of hash collision.

$$P_{h \in H}(h_i(X) = h_i(Y)) \leq p_1, \text{ if } S(X, Y) < \epsilon_1$$

$$P_{h \in H}(h_i(X) = h_i(Y)) \geq p_2, \text{ if } S(X, Y) > \epsilon_2$$

Given that,  $p_2 > p_1$ , while  $X, Y \in \mathbb{R}^n$ , and  $H = \{h : \mathbb{R}^n \rightarrow M\}$ , where  $M$  is the hashed metric space depends to similarity function defined by  $S$ ,  $i$  refers to the number of hash functions  $h$ .

### 3 Proposed Method

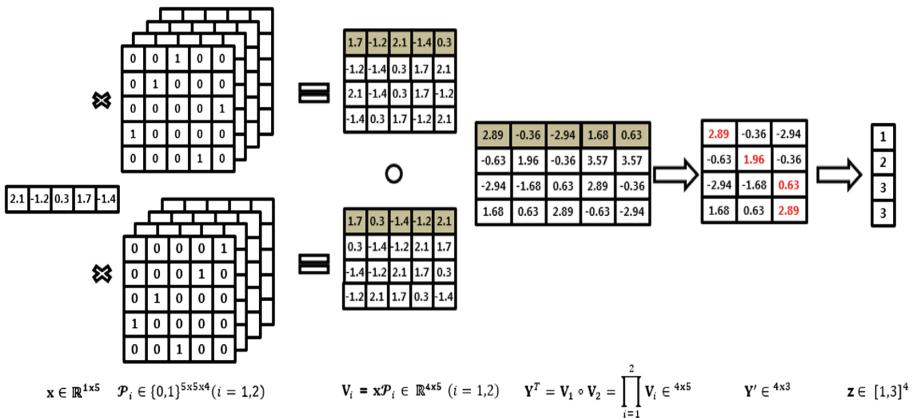
In this section, we present a ranking based hashing as a means of cancellable biometrics construct. Assume an input feature vector  $\mathbf{x} \in \mathbb{R}^n$ , the hashed code is generated according to the procedure as follows:

1. Generate  $p$  number of 3D permutation arrays that made by stacking up  $m \times n \times n$  permutation matrices  $\mathcal{P}_i \in \{0,1\}^{n \times n \times m}$ ,  $i = 1, \dots, p$  where  $m$  is the desired length of resulting hashed code and  $m > n$ . Note a permutation matrix is a square binary matrix that has exactly one entry of 1 in each row and each column and 0 s elsewhere.  $\mathcal{P}_i$  is user-specific.
2. Multiplying  $\mathbf{x}$  to  $\mathcal{P}_i$ , yield a matrix,  $\mathbf{V}_i = \mathbf{x}\mathcal{P}_i \in \mathbb{R}^{n \times m}$ ,  $i = 1, \dots, p$ .
3. Perform Hadamard product (element-wise product) yields  $\mathbf{H} = \prod_{i=1}^p \mathbf{V}_i \in \mathbb{R}^{n \times m}$  where  $p$  is Hadamard product order.
4. Discard last  $n - k$  column vectors from  $\mathbf{H}$ , yields  $\mathbf{H}' \in \mathbb{R}^{k \times m}$ , where we named  $k$  as window size.
5. For each row vectors of  $\mathbf{H}'$  the *indices of the largest magnitude entry* are recorded and form a discrete hashed code,  $\mathbf{h} \in [1 \ k]^m$ .

Note the proposed ranking based hashing is not exactly identical to WTA hashing described in Sect. 2 in the sense that our method is reformulated into a mathematically equivalent non-iterative matrix form instead of WTA iterative algorithmic form. Furthermore, Hadamard product is introduced to enhance the security and privacy protection strength. Figure 1 illustrates the proposed ranking based hashing.

Similar to WTA hashing that follows LSH theory that strives to ensure two similar biometric vectors renders higher probability of match (collision) in the rank domain, and vice versa for the vectors that are far apart to each other. Indeed, each entry  $h_i$  in the hashed code  $\mathbf{h}$  (step 5) can be seen as a LSH projected instance of biometric vector  $\mathbf{x} \in \mathbb{R}^n$  i.e.  $h_i = h(\mathbf{x}) \in [1, k]$  where  $h(\mathbf{x}) \in \{j \mid \max_j(\text{trunc}_k(\prod_{i=1}^p \mathbf{x}\mathcal{P}_i))\}$ ,  $\mathcal{P}_i \in \{0,1\}^{n \times n \times 1}$ , index  $j$  and  $\text{trunc}_k()$  refers to a function that discards last  $m-k$  entries of a vector.

Suppose two ranking based hashed codes, enrolled  $\mathbf{h}^e$  and query  $\mathbf{h}^q$  generated from fingerprint vector  $\mathbf{x}^e$  and  $\mathbf{x}^q$  respectively, the probability of match can be calculated by counting the number of agreed position (i.e. indices of  $\mathbf{h}$ ) between  $\mathbf{h}^e$  and  $\mathbf{h}^q$  over  $m$  as



**Fig. 1.** Illustration of ranking based hashing with  $n = 5$ ,  $p = 2$ ,  $k = 3$  and  $m = 4$ . Note  $m > n$  in our experiment, yet the parameters ( $n = 5$ ,  $m = 4$ ) in figure 1 is only for illustration.

$P[h_i^e - h_j^q] = S(\mathbf{x}, \mathbf{y})$  for  $i = 1, \dots, m$ . As a WTA hashing instance,  $S(\mathbf{x}, \mathbf{y})$  represents the similarity measure between two hashed codes that corresponds to the rank correlation measurement (a type of ordinal measure [18, 20]) of  $\mathbf{x}^e$  and  $\mathbf{x}^q$ . This suggested the similarity would preserve as the hashed codes collision.

In the event of template compromised, a new hashed code can be re-issued by repeating the above 5-step procedure. The effectiveness of cancellability is experimentally verified in Sect. 5.3.

In real world scenario, the permutation seed is user-specific for cancellability. However, lost token/seed case should be primary attended, as it is closely associated to accuracy performance, security and privacy attacks [1, 19]. To evaluate the lost token scenario, our experiment is performed with same permutation seed for all subjects (presented in Sect. 4.2).

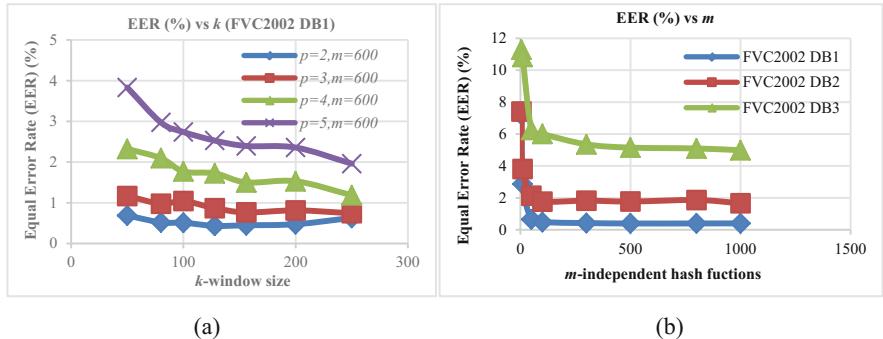
## 4 Experiments and Discussions

In this paper, a real-valued fixed-length fingerprint vector with size 299 that generated from MCC and Kernel Principal Component Analysis [21] is used as input to evaluate the proposed method. We refer the readers for the details about the fingerprint vector construction in [22]. The evaluations are conducted on six public fingerprint datasets, FVC2002 (DB1, DB2, DB3) [23] and FVC2004 (DB1, DB2, DB3) [23]. Each dataset consists of 100 users with 8 samples per user. In total, there are 800 ( $100 \times 8$ ) fingerprint images in each dataset. The performance accuracy of the proposed method is assessed using Equal Error Rate (EER) and the genuine-impostor distribution. Noted that since the random permutation is applied, to avoid the bias of single random permutation, the EERs is calculated by taking the average of EER repeated for 5 times. The fixed-length feature vector generated from fingerprint is described in [22].

For matching protocol, as described in [22], 1<sup>st</sup> to 3<sup>rd</sup> samples of each identity are used as training samples to generate the fingerprint vector; the rest samples (i.e. 4<sup>th</sup>–8<sup>th</sup>) of each identity are used in this experiment. There are totally 500 ( $100 \times 5$ ) samples used for experiment. Within this subset of data, The Fingerprint Verification Competition (FVC) [23] protocol is applied across the six data sets, which yields 1000 genuine matching scores and 4950 imposter matching scores for each data set.

#### 4.1 Effect of Window Size $k$ , Hadamard Product Order $p$ , and Number of Hashing Functions $m$

We first investigate the effect of window size  $k$  with respect to the performance in terms of EER. In this experiment,  $k$  is varied from 50, 80, 100, 128, 156, 200 to 250 with  $m = 600$ . The identical setting is repeated for  $p = [2, 3, 4, 5]$ . Figure 2(a) shows the curves of “EER (%) vs  $k$ ” for FVC2002 DB1. Note we repeat the same experiments for DB2 and DB3, but only DB 1 is shown as all experiments exhibit the same performance trend.



**Fig. 2.** (a) The curves of “EER (%) vs  $k$ ”; (b) The curves of “EER (%) vs  $m$ ”

We can observe that:

- (1) The EER drops gradually when larger  $k$  is applied and levels off when  $k$  becomes large. This is not a surprise as small  $k$  implies less feature components are taken into account, which leads to insufficient discriminability; while larger  $k$  indicates more salient features are included;
- (2) The smaller  $p$ , the lower EER. Step 3 described in Sect. 3 tells us that the Hadamard product is carried out by element-wise multiplying  $p$  permuted fingerprint vector in  $\mathbf{x}$  i.e.  $\bar{\mathbf{x}}(j) = \prod_{l=1}^p (\hat{\mathbf{x}}_l(j))$ . Such an operation heightens the hardness against inversion at the expense of introducing distortion in the product code. Thus, it is expected that the performance drops with large  $p$ . This also demonstrates the common trade-off exists in cancellable biometric scheme, namely performance-security trade-off.

We also examine the relation of the number of hashing functions  $m$  and EER. Evaluation has been carried out by increasing the  $m$  from 5, 10, 50, 100, 300, 500, 800 and 1000 while fixing  $k = 250$ , and  $p = 2$ . As expected, a better EER can be gained with respect to the increment of  $m$  and level off at large  $m$  as illustrated in Fig. 2(b). This performance pattern is expected as the large  $m$  increases the collision probability of two highly similar hashed codes and vice versa, which is theoretically assured by the WTA hashing.

## 4.2 Accuracy Performance Evaluation

In this section, the accuracy performance experiments on FVC2002 and FVC2004 using the best parameters found in the previous section is carried out. Table 1 presents the accuracy performance as well as comparisons with the baseline systems and BTP schemes. The accuracy performance of ranking based hashing gradually decreases in FVC2002 (DB1 and DB2) and FVC2004 DB1 while remains approximately 3%-5% of deterioration in the rest of data sets. Such deterioration is expected, as the discriminative features are likely to be permuted out of the  $k$ -window. However, the use of user-specific seed compensates the loss of discriminative features; thus, the accuracy in genuine-token case is comparable to its original vector counterpart [18] and MCC [9]. This suggests that the ranking based hashing demands higher discriminative features in order to preserve accuracy. Nevertheless, the proposed method mostly outperforms state-of-the-arts [10, 17, 24] in which same datasets and protocol are adopted.

**Table 1.** Performance accuracy and comparison.

FVC2002			FVC2004		
DB1	DB2	DB3	DB1	DB2	DB3
<i>Without template protection</i>					
MCC [9]					
0.60%	0.59%	3.91%	3.97%	5.22%	3.82%
Fixed-length representation [23]					
0.20%	0.19%	2.30%	4.70%	3.13%	2.80%
<i>With template protection</i>					
<b>Proposed (lost token case)</b>					
0.43%	2.10%	<b>6.60%</b>	<b>4.51%</b>	<b>8.02%</b>	<b>8.46%</b>
<b>Proposed (genuine token)</b>					
0.20%	0.88%	1.94%	0.44%	3.08%	2.91%
2P-MCC64,64 [10]					
3.3%	1.8%	7.8%	6.3%	–	–
Bloom filter [25]					
2.3%	1.8%	6.6%	13.4%	8.1%	9.7%
Wang and Hu [17]					
4%	3%	8.5%	–	–	–

## 5 Privacy and Security Analysis

In this section, we provide the privacy and security analysis that consists of (1) non-invertibility/irreversibility analysis; (2) brute force attack and false accept attack; (3) revocability; (4) non-linkability analysis.

### 5.1 Noninvertibility/Irreversibility Analysis

Non-invertibility/Irreversibility analysis, a privacy analysis, refers to the computational hardness in restoring the fingerprint vector from the hashed code with and without information associated to hashing algorithm.

Here, we assume the adversary manages retrieve the hashed codes and he knows well the hashing algorithm as well as the corresponding parameters (e.g.  $m$ ,  $k$ ,  $p$  and permutation seeds). We noted that the proposed ranking based hashing converts the real-valued fingerprint feature into the index value. Hence, it is reasonable to assume there is no clue for an adversary to guess the fingerprint vector information directly from the stolen hashed code alone or even with the parameters.

The only way for the adversary to attack is to guess the real-value features directly. In the worst case, the adversary learns the minimum and maximum values of the feature components. Let's take FVC2002 DB1 as an example, the minimum and maximum values of the feature components are  $-0.2504$  and  $0.2132$  respectively. The adversary has to examine from  $-0.2504$ ,  $-0.2503$ ,  $-0.2502$  and so on, until  $0.2132$ . Thus, there are 4636 possibilities. In our implementation, the precision is fixed at four decimal digits, the possibility of guessing a single feature component of fingerprint vector requires 4636 attempts ( $\approx 2^{12}$ ) Thus, the 299 feature components of a fingerprint vector require around  $2^{12 \times 299} = 2^{3588}$  attempts in total. The possibilities to correctly guess a single and entire feature components are given in Table 2. Obviously, such combinations are computationally infeasible.

**Table 2.** Complexity to invert single and entire feature components

Databases	Min value	Max value	Possibilities for single feature component	Total possibilities for entire feature
FVC2002 DB1	-0.2504	0.2132	$4636 \approx 2^{12}$	$2^{12 \times 299} = 2^{3588}$
FVC2002 DB2	-0.2409	0.2484	$4893 \approx 2^{12}$	$2^{12 \times 299} = 2^{3588}$
FVC2002 DB3	-0.1919	0.2372	$4291 \approx 2^{12}$	$2^{12 \times 299} = 2^{3588}$

**Table 3.** False accept attack complexity

Databases	$\tau$	$m$	$\tau \times m$	$k$	Minimum attack complexity
FVC2002 DB1	0.11	600	66	$128 = 2^7$	$(2^7)^{66} = 2^{462}$
FVC2002 DB2	0.08	600	48	$250 \approx 2^8$	$(2^8)^{48} = 2^{384}$
FVC2002 DB3	0.05	600	30	$250 \approx 2^8$	$(2^8)^{30} = 2^{240}$

## 5.2 Brute Force Attack and False Accept Attack

Brute-force attack is an instance of security attacks, which meant to gain the illegitimate access, with feasible attack complexity to the biometric system by means of the randomly generated hashed code. We quantitatively analyze the required complexity to break the proposed method. For the realization, assume that the optimal parameters for the best performance are set to  $m = 600$  and  $k = 128$ . Since the indices of hashed code taking a value between 1 and 128, the guess complexity for each entry is  $k = 128 = 2^7$  and thus 600 entries require  $2^{4200}$  attempts that are far beyond to reach by the present computing facility.

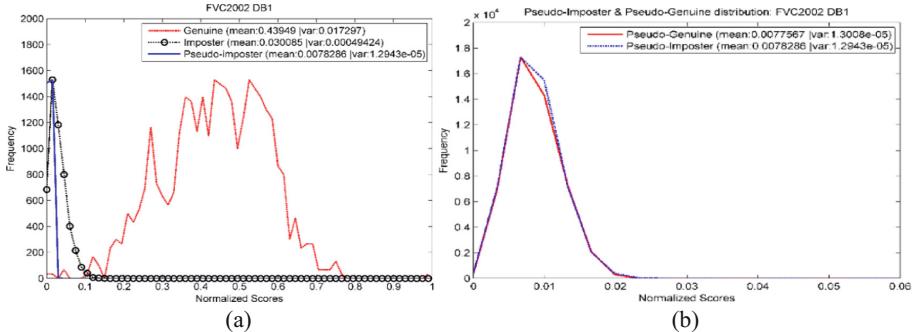
On the other hand, unlike the brute force attack, false accept attack (dictionary attack) may requires far less number of attempts to gain illegitimate access. In fact, biometric systems make decision based on the system threshold value, the access would be granted as long as the matching score succeeds the pre-defined threshold  $\tau$ , which can significantly reduce the attack effort.

Let we take the best performing parameters from the FVC2002DB1 experiments in Sect. 4.1, i.e.  $m = 600$  and  $k = 128$ , the decision threshold  $\tau$  observed at this setting is 0.11. Hence, the minimum number of agreed (collided) entries in the hashed codes pair for successful access is mere  $\tau \times m = 0.11 \times 600 = 66$ . The window size  $k$  indicates 128 possible indices values that is equivalent to  $2^7$  ( $= 2^{\log_2 k}$ ) guessing effort is required for an entry. Therefore, the false accept attack complexity can be estimated from  $(2^{\log_2 k})^{\tau \times m}$ . The complexity calculated is shown in Table 3, we observe that the attack complexity reduced to  $2^{462}$  compare to  $2^{4200}$  in brute force attack for FVC2002DB1 and the complexity reduction appears identical for the rest of testing data sets. However, we note that the reduced attack complexity is still favorably high to resist the false accept attack.

## 5.3 Revocability

Revocability is evaluated by conducting the experiment where 100 hashed codes of each fingerprint vector are generated with 100 sets distinct random permutation seeds, and then the first hashed code is matched with the other 100 hashed codes. The entire process is repeated and produces  $100 \times (5 \times 100) = 50000$  pseudo-impostor scores. The genuine, imposter, and pseudo-impostor distribution are computed with  $p = 2, k = 128, m = 600$  as depicted in Fig. 3(a).

Note that the numbers of scores are different for the imposter and pseudo-impostor matching. This is because in pseudo-impostor matching, we only focus on the matching scores between the first hashed code and the newly generated hashed code for each fingerprint vector (same user). From Fig. 3, a large degree of overlapping occurs between the imposter and pseudo-impostor distributions. This implies the newly generated hashed codes with the given 100 random permutation seeds are distinctive despite it is generated from the identical fingerprint vector. In terms of verification performance, we obtain EER = 0.16% in which intersection of genuine and pseudo-impostor distribution is taken. This verifies that the proposed method satisfies the revocability property requirement.



**Fig. 3.** (a) The genuine, imposter, and pseudo-impostor distribution:  $p = 2$ ,  $k = 128$ ,  $m = 600$  on FVC2002 DB1; (b) Pseudo-impostor & pseudo-genuine distribution on FVC2002 DB1.

#### 5.4 Non-linkability

To examine the non-linkability of our scheme, we introduce the *pseudo-genuine scores*, which refer to the matching scores between two hashed codes under different fingerprint's vector of the same individual (using different permutation seeds). This resembles the genuine matching that yields 1000 scores. Recall the pseudo-impostor score used under Sect. 5.2, when the pseudo-impostor and pseudo-genuine distributions are overlapped, it implies that the hashed codes generated from the same user or from the others are not differentiable. On the contrary, if both distributions are separated far apart, this offers the advantages to an adversary to differentiate the hashed code from the identical individual. The difficulty in differentiating the hashed codes contributed to the non-linkability. Figure 3(b) illustrates the pseudo-impostor and pseudo-genuine distribution plot where the pseudo-impostor and pseudo-genuine distribution are largely overlapped hence, supports the non-linkability property.

## 6 Conclusion

In this paper, we presented a cancellable ranking based hashing for fingerprint template protection. The hashed codes can largely preserve accuracy performance with respect to its original counterparts thanks to the nice property that inherited from WTA hashing. The scheme is also shown to satisfy both non-linkability and revocability criteria. The hashed code is strongly resilient against the non-invertibility analysis due to its indices representation that carry no information about original biometric data. We also demonstrate its resilience to brute force and dictionary attacks. Our future work consists of two directions. The first direction is to extend the work to unordered variable-sized representation such as fingerprint minutiae. The second direction is to integrate with biometric cryptosystem primitives such as Fuzzy Vault, Fuzzy Commitment etc., which would be a strong complement for cryptographic keys generation and protection purposes.

**Acknowledgement.** This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2016-0-00097, Development of Biometrics-based Key Infrastructure Technology for On-line Identification).

## References

1. Teoh, A.B.J., Goh, A., Ngo, D.C.L.: Random multispace quantization as an analytic mechanism for BioHashing of biometric and random identity inputs. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(12), 1892–1901 (2006)
2. Jain, A.K., Nandakumar, K., Nagar, A.: Biometric template security. *EURASIP J. Adv. Sig. Process.* **2008**, 579416 (2008)
3. Ratha, N.K., Chikkerur, S., Connell, J.H., Bolle, R.M.: Generating cancelable fingerprint templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(4), 561–572 (2007)
4. Rathgeb, C., Uhl, A.: A survey on biometric cryptosystems and cancelable biometrics. *EURASIP J. Inf. Secur.* **2011**(1), 3 (2011)
5. Patel, V.M., Ratha, N.K., Chellappa, R.: Cancelable biometrics: a review. *IEEE Sig. Process. Mag.* **32**(5), 54–65 (2015)
6. Natgunanathan, I., Mehmood, A., Xiang, Y., Beliakov, G., Yearwood, J.: Protection of privacy in biometric data. *IEEE Access* **4**, 880–892 (2016)
7. Nandakumar, K., Jain, A.K.: Biometric template protection schemes: bridging the performance gap between theory and practice. *IEEE Sig. Process. Mag.* **32**(5), 88–100 (2015)
8. Ferrara, M., Maltoni, D., Cappelli, R.: Noninvertible minutia cylinder-code representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **7**(6), 1727–1737 (2012)
9. Cappelli, R., Ferrara, M., Maltoni, D.: Minutia cylinder-code: a new representation and matching technique for fingerprint recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(12), 2128–2141 (2010)
10. Ferrara, M., Maltoni, D., Cappelli, R.: A two-factor protection scheme for MCC fingerprint templates. In: 2014 International Conference of the Biometrics Special Interest Group, pp. 1–8 (2014)
11. Rathgeb, C., Breitinger, F., Busch, C., Baier, H.: On application of bloom filters to iris biometrics. *IET Biom.* **3**(4), 207–218 (2014)
12. Gomez-Barrero, M., Rathgeb, C., Galbally, J., Fierrez, J., Busch, C.: Protected facial biometric templates based on local gabor patterns and adaptive bloom filters. In: ICPR, pp. 4483–4488, August 2014
13. Li, G., Yang, B., Rathgeb, C., Busch, C.: Towards generating protected fingerprint templates based on bloom filters. In: 2015 International Workshop on Biometrics and Forensics, pp. 1–6 (2015)
14. Hermans, J., Mennink, B., Peeters, R.: When a bloom filter becomes a doom filter: security assessment of a novel iris biometric template protection system. In: Proceedings of the Special Interest Group on Biometrics, Darmstadt, pp. 1–6, September 2014
15. Bringer, J., Morel, C., Rathgeb, C.: Security analysis of bloom filter-based iris biometric template protection. In: Proceedings of International Conference on Biometrics (ICB), pp. 527–534 (2015)
16. Sandhya, M., Prasad, M.V.K.: k-Nearest Neighborhood Structure (k-NNS) based alignment-free method for fingerprint template protection. In: ICB 2015, pp. 386–393 (2015)
17. Wang, S., Hu, J.: A blind system identification approach to cancelable fingerprint templates. *Pattern Recognit.* **54**, 14–22 (2016)

18. Yagnik, J., Strelow, D., Ross, D.A., Lin, R.-S.: The power of comparative reasoning. In: IEEE ICCV, pp. 2431–2438 (2011)
19. Nagar, A.: Biometric template security. Ph.D. dissertation, Department of Computer Science and Engineering, Michigan State University (2012)
20. Bhat, D., Nayar, S.: Ordinal measures for visual correspondence. In: Proceedings of CVPR, pp. 351–357 (1996)
21. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **10**, 1299–1319 (1998)
22. Jin, Z., Lim, M.H., Teoh, A.B.J., Goi, B.M., Tay, Y.H.: Generating fixed-length representation from minutiae using kernel methods for fingerprint authentication. *IEEE Trans. Syst. Man Cybern. Syst.* **40**(10), 1415–1428 (2016)
23. BioLab: FVC2002, FVC2004. <http://bias.csr.unibo.it>
24. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proceedings of 30th Symposium on Theory of Computing (1998)
25. Abe, N., Yamada, S., Shinzaki, T.: Irreversible fingerprint template using Minutiae Relation Code with Bloom Filter. In: IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS), Arlington, pp. 1–7 (2015)

## Author Index

- Aboutorab, Neda 1  
Alepis, Efthimios 14  
Alipio, Melchizedek I. 217  
Alzahrami, Eidah J. 247  
Aziz, Izzatdin A. 247
- Baccour, Emna 188  
Bazai, Sibghat Ullah 364  
Bouras, Abdelaziz 178
- Cai, Qing 277  
Chang, Yoon Seok 231  
Chaudhry, Junaid 291  
Chen, Chi 66  
Chin, Ji-Jian 150  
Cho, Sangrae 378  
Creech, Gideon 353
- Deepthi, Kakumani K. C. 324  
Dou, Wanchun 277  
Du, Xiaojiang 313, 339
- Foufou, Sebti 188  
Fu, Jing 79
- Gao, Feng 313, 339  
Gasmi, Houssem 178  
Ghemri, Fadi 178  
Goscinski, Andrzej M. 247  
Gouissem, Ala 188  
Guo, Ying 164
- Haider, Waqas 137  
Hamila, Ridha 188  
Hu, Jiankun 56, 66, 116, 127, 137, 164, 291  
Hwang, Jung Yeon 378
- Ibrahim, Ahmed 291  
Iffländer, Lukas 262
- Jang-Jaccard, Julian 364  
Jin, Zhe 378
- Kim, Soohyung 378  
Koroniotis, Nickolaos 30  
Kounev, Samuel 262
- Lai, Yen-Lung 378  
Lee, Gaemyoung 231  
Li, Fei 164  
Li, Yandong 313, 339  
Liu, Sheng 313, 339  
Liu, Zifan 277  
Loquias, Rizza T. 87
- Ma, Jixin 299  
Ma, Shunan 79  
Marciano, Joel Joseph S. 87  
Marsden, Thomas 353  
Metter, Christopher 262  
Moustafa, Nour 30, 137, 353  
Mubarak, Khalid 231  
Mulerikkal, Jaison Paul 45
- Ng, Tiong-Sik 150
- Palamakumbura, Sudharaka 203  
Patsakis, Constantinos 14  
Pedrasa, Jhoanna Rhodette I. 87
- Qin, Jing 299  
Quinton, Ben 1
- Salih, Yasir 231  
Sarker, Ruhul 116  
Shemaili, Mouza Ahmed Bani 231  
Shen, Meng 313, 339  
Shen, Peisong 66  
Shi, Shi 79  
Simsim, Mohammed 231  
Singh, Kunwar 324  
Sitnikova, Elena 30, 353  
Slay, Jill 30  
Sona, C. P. 45  
Sun, Jiameng 299

- Tan, Syh-Yuan 150  
Tari, Zahir 188, 247  
Teoh, Andrew Beng Jin 378  
Tian, Xue 66  
Tiglao, Nestor Michael C. 87, 217  
Tran, Nam Nhat 116  
Tran, Quang Nhat 101  
Tran, Yen Hong 101  
Tran-Gia, Phuoc 262  
  
Usefi, Hamid 203  
  
Valli, Craig 291  
  
Wamser, Florian 262  
Wang, Ruili 364  
Wang, Song 277, 291  
  
Xu, Xiaolong 277
- Yang, Jing 313  
Yang, Tengfei 66  
Yang, Wencheng 291  
Yeun, Chan Yeob 231  
Yeun, Hyun Ku 231  
Yin, Shu 313, 339  
Yin, Xuefei 56, 127  
Yu, Shui 277  
  
Zafar, Basim 231  
Zemerly, Mohamed Jamal 231  
Zhang, Chuan 313  
Zheng, Baokun 313, 339  
Zheng, Guanglou 291  
Zhou, Zhaohua 79  
Zhu, Binrui 299  
Zhu, Liehuang 313, 339  
Zhu, Yanming 56, 127  
Zomaya, Albert Y. 188