

# Rethinking Authentication on Smart Mobile Devices 2020

Lead Guest Editor: Ding Wang

Guest Editors: Kun Sun and Qi Jiang





---

# **Rethinking Authentication on Smart Mobile Devices 2020**

Wireless Communications and Mobile Computing

---

# **Rethinking Authentication on Smart Mobile Devices 2020**

Lead Guest Editor: Ding Wang

Guest Editors: Kun Sun and Qi Jiang



---



Copyright © 2022 Hindawi Limited. All rights reserved.

This is a special issue published in “Wireless Communications and Mobile Computing.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

# Chief Editor

































Zhipeng Cai , USA

## Associate Editors

Ke Guan , China  
Jaime Lloret , Spain  
Maode Ma , Singapore

## Academic Editors

Muhammad Inam Abbasi, Malaysia  
Ghufran Ahmed , Pakistan  
Hamza Mohammed Ridha Al-Khafaji ,  
Iraq  
Abdullah Alamoodi , Malaysia  
Marica Amadeo, Italy  
Sandhya Aneja, USA  
Mohd Dilshad Ansari, India  
Eva Antonino-Daviu , Spain  
Mehmet Emin Aydin, United Kingdom  
Parameshchhari B. D. , India  
Kalapaveen Bagadi , India  
Ashish Bagwari , India  
Dr. Abdul Basit , Pakistan  
Alessandro Bazzi , Italy  
Zdenek Becvar , Czech Republic  
Nabil Benamar , Morocco  
Olivier Berder, France  
Petros S. Bithas, Greece  
Dario Bruneo , Italy  
Jun Cai, Canada  
Xuesong Cai, Denmark  
Gerardo Canfora , Italy  
Rolando Carrasco, United Kingdom  
Vicente Casares-Giner , Spain  
Brijesh Chaurasia, India  
Lin Chen , France  
Xianfu Chen , Finland  
Hui Cheng , United Kingdom  
Hsin-Hung Cho, Taiwan  
Ernestina Cianca , Italy  
Marta Cimitile , Italy  
Riccardo Colella , Italy  
Mario Collotta , Italy  
Massimo Condoluci , Sweden  
Antonino Crivello , Italy  
Antonio De Domenico , France  
Floriano De Rango , Italy

Antonio De la Oliva , Spain  
Margot Deruyck, Belgium  
Liang Dong , USA  
Praveen Kumar Donta, Austria  
Zhuojun Duan, USA  
Mohammed El-Hajjar , United Kingdom  
Oscar Esparza , Spain  
Maria Fazio , Italy  
Mauro Femminella , Italy  
Manuel Fernandez-Veiga , Spain  
Gianluigi Ferrari , Italy  
Luca Foschini , Italy  
Alexandros G. Fragkiadakis , Greece  
Ivan Ganchev , Bulgaria  
Óscar García, Spain  
Manuel García Sánchez , Spain  
L. J. García Villalba , Spain  
Miguel Garcia-Pineda , Spain  
Piedad Garrido , Spain  
Michele Girolami, Italy  
Mariusz Glabowski , Poland  
Carles Gomez , Spain  
Antonio Guerrieri , Italy  
Barbara Guidi , Italy  
Rami Hamdi, Qatar  
Tao Han, USA  
Sherief Hashima , Egypt  
Mahmoud Hassaballah , Egypt  
Yejun He , China  
Yixin He, China  
Andrej Hrovat , Slovenia  
Chunqiang Hu , China  
Xuexian Hu , China  
Zhenghua Huang , China  
Xiaohong Jiang , Japan  
Vicente Julian , Spain  
Rajesh Kaluri , India  
Dimitrios Katsaros, Greece  
Muhammad Asghar Khan, Pakistan  
Rahim Khan , Pakistan  
Ahmed Khattab, Egypt  
Hasan Ali Khattak, Pakistan  
Mario Kolberg , United Kingdom  
Meet Kumari, India  
Wen-Cheng Lai , Taiwan

Jose M. Lanza-Gutierrez, Spain  
Pavlos I. Lazaridis , United Kingdom  
Kim-Hung Le , Vietnam  
Tuan Anh Le , United Kingdom  
Xianfu Lei, China  
Jianfeng Li , China  
Xiangxue Li , China  
Yaguang Lin , China  
Zhi Lin , China  
Liu Liu , China  
Mingqian Liu , China  
Zhi Liu, Japan  
Miguel López-Benítez , United Kingdom  
Chuanwen Luo , China  
Lu Lv, China  
Basem M. ElHalawany , Egypt  
Imadeldin Mahgoub , USA  
Rajesh Manoharan , India  
Davide Mattera , Italy  
Michael McGuire , Canada  
Weizhi Meng , Denmark  
Klaus Moessner , United Kingdom  
Simone Morosi , Italy  
Amrit Mukherjee, Czech Republic  
Shahid Mumtaz , Portugal  
Giovanni Nardini , Italy  
Tuan M. Nguyen , Vietnam  
Petros Nicolitidis , Greece  
Rajendran Parthiban , Malaysia  
Giovanni Pau , Italy  
Matteo Petracca , Italy  
Marco Picone , Italy  
Daniele Pinchera , Italy  
Giuseppe Piro , Italy  
Javier Prieto , Spain  
Umair Rafique, Finland  
Maheswar Rajagopal , India  
Sujan Rajbhandari , United Kingdom  
Rajib Rana, Australia  
Luca Reggiani , Italy  
Daniel G. Reina , Spain  
Bo Rong , Canada  
Mangal Sain , Republic of Korea  
Praneet Saurabh , India

Hans Schotten, Germany  
Patrick Seeling , USA  
Muhammad Shafiq , China  
Zaffar Ahmed Shaikh , Pakistan  
Vishal Sharma , United Kingdom  
Kaize Shi , Australia  
Chakchai So-In, Thailand  
Enrique Stevens-Navarro , Mexico  
Sangeetha Subbaraj , India  
Tien-Wen Sung, Taiwan  
Suhua Tang , Japan  
Pan Tang , China  
Pierre-Martin Tardif , Canada  
Sreenath Reddy Thummaluru, India  
Tran Trung Duy , Vietnam  
Fan-Hsun Tseng, Taiwan  
S Velliangiri , India  
Quoc-Tuan Vien , United Kingdom  
Enrico M. Vitucci , Italy  
Shaohua Wan , China  
Dawei Wang, China  
Huaqun Wang , China  
Pengfei Wang , China  
Dapeng Wu , China  
Huaming Wu , China  
Ding Xu , China  
YAN YAO , China  
Jie Yang, USA  
Long Yang , China  
Qiang Ye , Canada  
Changyan Yi , China  
Ya-Ju Yu , Taiwan  
Marat V. Yuldashev , Finland  
Sherali Zeadally, USA  
Hong-Hai Zhang, USA  
Jiliang Zhang, China  
Lei Zhang, Spain  
Wence Zhang , China  
Yushu Zhang, China  
Kechen Zheng, China  
Fuhui Zhou , USA  
Meiling Zhu, United Kingdom  
Zhengyu Zhu , China




# Contents

## **PUF-Assisted Lightweight Group Authentication and Key Agreement Protocol in Smart Home**

Yandong Xia, Rongxin Qi , Sai Ji, Jian Shen , Tiantian Miao, and Huaqun Wang




Research Article (15 pages), Article ID 8865158, Volume 2022 (2022)

## **Provably Secure ECC-Based Three-Factor Authentication Scheme for Mobile Cloud Computing with Offline Registration Centre**

Hongwei Luo , Feifei Wang , and Guoai Xu 


Research Article (12 pages), Article ID 8848032, Volume 2021 (2021)

## **Security Analysis of Out-of-Band Device Pairing Protocols: A Survey**

Sameh Khalfaoui , Jean Leneutre , Arthur Villard, Jingxuan Ma, and Pascal Urien 





Review Article (30 pages), Article ID 8887472, Volume 2021 (2021)

## **Determining the Image Base of Smart Device Firmware for Security Analysis**

Ruijin Zhu , Baofeng Zhang, Yu-an Tan, Jinmiao Wang , and Yueliang Wan

Research Article (12 pages), Article ID 8899193, Volume 2020 (2020)

## **Study on Security and Privacy in 5G-Enabled Applications**

Qin Qiu , Shenglan Liu , Sijia Xu , and Shengquan Yu 






Research Article (15 pages), Article ID 8856683, Volume 2020 (2020)

## **Two-Round Password-Based Authenticated Key Exchange from Lattices**

Anqi Yin , Yuanbo Guo , Yuanming Song , Tongzhou Qu , and Chen Fang 

Research Article (13 pages), Article ID 8893628, Volume 2020 (2020)

## **Privacy-Preserving Graph Operations for Mobile Authentication**

Peng Li , Fucai Zhou , Zifeng Xu , Yuxi Li , and Jian Xu 



Research Article (13 pages), Article ID 8859213, Volume 2020 (2020)

## **Security Analysis on “Anonymous Authentication Scheme for Smart Home Environment with Provable Security”**

Meijia Xu , Qiyong Dong , Mai Zhou , Chenyu Wang , and Yangyang Liu 


Research Article (4 pages), Article ID 8838363, Volume 2020 (2020)

## **From Hardware to Operating System: A Static Measurement Method of Android System Based on TrustZone**

Xinhong Hei, Wen Gao , Yichuan Wang , Lei Zhu, and Wenjiang Ji



Research Article (13 pages), Article ID 8816023, Volume 2020 (2020)

## **An Efficient Anonymous Authentication Scheme for Mobile Pay-TV Systems**

Yuting Li, Qingfeng Cheng , and Jinzheng Cao

Research Article (12 pages), Article ID 8850083, Volume 2020 (2020)

## **Authenticator Rebinding Attack of the UAF Protocol on Mobile Devices**

Hui Li , Xuesong Pan , Xinluo Wang, Haonan Feng, and Chengjie Shi

Research Article (14 pages), Article ID 8819790, Volume 2020 (2020)

## Research Article

# PUF-Assisted Lightweight Group Authentication and Key Agreement Protocol in Smart Home

Yandong Xia,<sup>1,2</sup> Rongxin Qi ,<sup>1,2</sup> Sai Ji,<sup>1,2</sup> Jian Shen ,<sup>1,2,3</sup> Tiantian Miao,<sup>1,2</sup>  
and Huaqun Wang<sup>4</sup>

<sup>1</sup>Nanjing University of Information Science and Technology, Nanjing, China

<sup>2</sup>Jiangsu Engineering Center of Network Monitoring, Nanjing, China

<sup>3</sup>Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, China

<sup>4</sup>Jiangsu Key Laboratory of Big Data Security and Intelligent Processing,  
Nanjing University of Posts and Telecommunications, China

Correspondence should be addressed to Jian Shen; s\_shenjian@126.com

Received 11 August 2020; Revised 18 January 2021; Accepted 25 February 2022; Published 24 March 2022

Academic Editor: Yin Zhang

Copyright © 2022 Yandong Xia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Various IoT-based applications such as smart home, intelligent medical, and VANETs have been put into practical utilization. The smart home is one of the most concerned environments, allowing users to access and control smart devices via the public network remotely. The smart home can provide many intelligent services for users through these smart devices. To securely access devices and obtain collected data over the public network, multifactor authentication protocols for smart home have gained wide attention. However, most of these protocols cannot withstand impersonation attack, smart device lost attack, privileged-insider attack, smart card lost attack, and so on. Besides, high communication and computational costs weaken the system performance, which leads to most authentication protocols are not suitable for resource-constrained smart devices. To mitigate the aforementioned drawbacks, we proposed a PUF-assisted lightweight group authentication and key agreement protocol to implement secure access to multiple devices in the smart home simultaneously using the Chinese Remainder Theorem and secret sharing technique. Our protocol also utilizes physical unclonable function (PUF) and fuzzy extractor technique to extract the digital fingerprint of the smart devices, which can uniquely validate smart devices and protect the secrets stored in their memory. Our protocol can support various security features and withstand the many well-known attacks in the smart home. The performance analysis indicates that the proposed protocol can efficiently reduce communication/computational costs when the user simultaneously accesses multiple devices.

## 1. Introduction

With the rapid development of the Internet of Things (IoT) technology, various IoT-based applications such as smart home, intelligent medical, and VANETs have emerged. In these applications, the smart home has gained wide attention in recent years due to its convenience, efficiency, and other properties, providing basic and practical home control services for users. The smart home is a dwelling that connects major appliances and services and permits them to be accessed via the public network [1]. In most existing schemes, the smart home is usually composed of user equip-

ment (e.g., smartphone), home gateway (HG), and lots of smart devices (e.g., surveillance camera, lighting controller, and temperature sensors) [2]. The smart devices are interconnected to collect the data in the smart home and interact with users via the public network. HG acts as the communication medium between the user and smart devices.

Smart devices are generally easy to suffer from various attacks such as impersonation attack, physical device lost attack, and privileged-insider attack during the execution of the protocol. Once these devices are broken, user privacy will be compromised. For example, unauthorized users may access the surveillance cameras and control



them to monitor smart home residents. In addition, most of these IoT devices such as sensors have limited resources to execute complex computational operations [3, 4]. In recent years, many Elliptic Curve Cryptography- (ECC-) based schemes [5, 6] have been proposed to enhance authentication security. However, these schemes generally require to perform complex computational operations, which are not suitable for resource-constrained devices. Some schemes also cannot provide most security features and functionalities such as user anonymity, perfect forward secrecy, and dynamic device addition. To solve the security and privacy issues in IoT environments, a large number of authentication schemes have been proposed [7–9]. In most of the existing schemes, the computational and communication costs are too high to be suitable for resource-constrained [8] devices. If the user wants to access multiple smart devices simultaneously, it is necessary to verify the authenticity of user identity frequently and send access requests to correspond with smart devices in a short time, which may lead to network delay and even congestion. Therefore, it is crucial to design an efficient and lightweight authentication scheme to establish the secure session key between the user and smart devices in the smart home. Group authentication schemes are put forward to solve aforementioned issues. Group authentication schemes based on secret sharing can authenticate multiple smart devices belonging to the same group simultaneously.

Besides, the traditional read-only memory- (ROM-) based authentication techniques have the characteristic of expensive power consumption and nonvolatile memory, which are vulnerable to external attacks [10]. Physical unclonable function is a promising hardware primitive that can be utilized for lightweight authentication and secret key storage, which extracts the unique physical property from the integrated circuits (IC) [11]. Each IC has different physical characteristics even if they are identical in function. The secrets derived from IC through PUF are actually different due to the variability in manufacturing. PUF can handle the inherent weaknesses successfully existing in the traditional ROM-based authentication techniques. PUF technique can be utilized to distinguish the smart devices and prevent them from being attacked, cloned, and forged by the adversary. However, changes in the environment around smart devices may affect the digital circuit, which leads to errors in the output of the PUF function. In order to improve the fault tolerance rate of the PUF function, the fuzzy extractor has been widely used to correct errors in the PUF function [12].

Considering the security of the parameters stored in the smart devices, PUF is utilized to prevent stolen device attack. PUF can be utilized to assist smart devices to generate a biometric key, which efficiently protects the security smart devices [12]. Therefore, we propose a PUF-assisted lightweight group authentication and key agreement protocol in the smart home. Our protocol supports many well-known features such as untraceability, user anonymity, and forward secrecy. The smart devices are allowed to join or leave the group dynamically.

### 1.1. Our Contributions

- (i) A PUF-assisted lightweight group authentication and key agreement protocol in the smart home is presented in our paper. Our protocol is suitable for the resource-constrained smart devices only using lightweight operation and symmetric cryptography. The secret sharing technique and Chinese Remainder Theorem are utilized to establish the group session key between the user and smart devices
- (ii) The security of our protocol is proved under the widespread ROR model [13]. The formal security analysis shows that our protocol is semantically secure. Other discussions on security show that the proposed protocol can guarantee many security features such as untraceability and user anonymity and also can withstand most known attacks
- (iii) The dynamic joining and leaving of smart devices from deployed network are both supported by the proposed protocol. The illegitimate smart devices fail to attain the group key without the secret share. The new smart device just registers itself before joining the deployed network
- (iv) The physical security of smart devices is guaranteed by physical unclonable function technology. The output of PUF depends on the physical fingerprint of the physical device. PUF has the characteristics of tamper-resistant, unclonability, and unpredictability
- (v) The issue of repeated authentication of the same user who accesses the multiple smart devices simultaneously is solved. The performance analysis indicates that the protocol effectively reduces resource costs compared with other protocols

### 1.2. Related Work

*1.2.1. Authentication.* Smart home allows the authorized users to remotely access devices and obtain information collected by these devices. To address security and privacy issues in IoT, a large number of researchers [14–16] have studied many authentication schemes for the smart home.

In 2011, Vaidya et al. proposed a novel authentication and key establishment mechanism based on ECC. Although their scheme satisfies more security requirements compared to previous schemes, their scheme is not suitable for resource-constrained home area networks. Therefore, many schemes focus on providing more security features while they are not suitable for resource-constrained devices. To solve communication security issues in WSNs, Xue et al. [14] utilized temporary credentials to implement authentication between the user and sensing nodes for WSNs in 2013. Their scheme is lightweight to be suitable for the sensing nodes using hash function and bit-wise XOR operations. However, He et al. [15] thought their scheme fails to resist offline password guessing attack, impersonation attack, and

tampering attack. In 2013, He et al. [17] proposed an improved authentication scheme that overcomes the security threats in Xue's scheme and only increases little computational cost. In 2014, Turkanovic et al. [17] focused on a scenario where the user accessing a single targeted sensor in WSNs does not need to interact with HG. Meanwhile, Kalra and Sood [18] found that Xue's scheme is vulnerable to smartcard lost attack. Kalra and Sood [18] proposed a novel authentication scheme based on password and smartcard, which can resist most known attacks and has a lower cost than other schemes. However, their scheme does not consider resisting sensing node capturing attack and privileged-insider attack. In 2018, Shen et al. [19] adopted the cloud to enhance the capabilities of devices and established a lightweight authentication scheme without certificates for WBANs.

The devices in the IoT environment have similar features to the sensing nodes in traditional WSNs. Due to the heterogeneity and dynamics of IoT devices, the higher security and privacy requirements need to be satisfied in the IoT environment. Kumar et al. [16] proposed an anonymous authentication framework for smart home only using hash function and symmetric cryptography. Kumar et al. firstly considered the features of anonymity and unlinkability for smart home, and their scheme can resist many known attacks. Challa et al. [20] proposed a novel signature-based authenticated key establishment scheme for the generic IoT environment. The user can not only communicate with smart devices but also with other users through HG. In 2018, Srinivas et al. [21] proposed an anonymous three-factor authentication and key agreement scheme which supports credentials update, user revocation, and new devices addition. However, Gope et al. [22] thought the sensitive information stored in the memory of smart devices may be compromised to the adversary by the side-channel attack. The adversary then obtains the sensitive information and traces all the access users in previous communications. Besides, most smart devices are not tamper-evident so that the adversary can intercept the communication messages and impersonate legitimate devices.

*1.2.2. Group Authentication.* The concept of group authentication is proposed to implement identity authentication among group members at a time. Many group-based authentication schemes are also proposed to improve the efficiency of group communication. In 2013, Harn [23] and Liu et al. [24] both proposed an improved group authentication protocol for group-oriented applications based on secret sharing. In 2016, Li et al. [25] thought that Harn's protocol fails to support key agreement during the authentication process and cannot resist replay attack and man-in-middle attack. They proposed an improved group authentication and key agreement protocol for MTC in LTE-A networks, which supports dynamical policy updating and provides strong security properties compared to previous work. In 2019, Cui et al. [26] proposed an efficient signature-based group authentication scheme for vehicular ad hoc networks (VANETs). RSU can efficiently update the group key generated by two hash chains to exclude malicious vehicles from

the group. In 2020, Zhang and Lee [27] provided an efficient group authentication scheme based on the group signature technique, which protects the integrity of blockchain-based mobile-edge computing (BMEC). In this paper, we propose a secure and efficient group authentication protocol for smart home based on the PUF and secret sharing technique. Currently, most of these protocols cannot withstand smart device lost attack and smart card lost attack. Besides, high communication and computational cost leads to most authentication protocols are not suitable for resource-constrained smart devices.

*1.2.3. PUF Technology.* Recently, PUF technology is introduced to resist the aforesaid issues. Most existing authentication protocols are designed based on tamper-evident PUF [28–35] to prevent the physical attack. Wallrabenstein [28] proposed an ideal PUF-based authentication protocol to provide cost-effective tamper resistance for resource-constrained devices in IoT, which minimizes the probability of private key disclosure. To resist denial and masquerading attacks, Chatterjee et al. [31] used PUF's response to replace the public identity string used for message encryption and disabled the public key generator in the scheme, allowing the receiving node to generate its own public and private keys and the server to verify the public key. In order to solve the problems of man-in-the middle attack and replay attack under DY security model, Braeken [32] used elliptic curve addition and multiplication to replace bilinear pair operation and realized identity-based authentication. Chatterjee et al. [33] combined IBE, PUF, and message authentication code to propose a low-power, low-latency authentication, and key agreement protocol that solves the database storage overhead and successfully defies man-in-the-middle attacks. Gope et al. [29] proposed a lightweight anonymous authentication protocol based on ideal PUF. They subsequently took the effects of noise on PUF into account and enhanced the authentication protocol to support noisy PUF. They utilized other prestored pseudo identities and challenge-response pairs to ensure the security of the protocol when suffering from DoS attacks. Furthermore, Tiplea and Hristea [30] pointed that most existing PUF-based authentication protocols cannot protect security and privacy in IoT under corruption with temporary state disclosure, while some important temporary variables are not protected by PUF. Therefore, they proposed a general method to protect the temporary variables and utilized it to fix the flaws existing in the previous PUF-based authentication protocols. Li and Liu [34] optimized the existing RFID authentication protocol based on double PUF. They proposed a protocol that can meet the untraceable, successfully resist desynchronization attacks and tag impersonation attacks, and has better security and privacy. PUF-based authentication schemes are threatened by powerful machine learning attacks. Chen et al. [35] show that the "availability" and "reliability" features of Shamir's secret sharing (SSS) can be applied to address the security issue. They presented a mutual authentication protocol where no response is exposed to the adversary and can avoid the use of cryptographic algorithms and error correcting codes. The current PUF-based

authentication protocol can resist internal attacks, but it is still affected by external environment, resulting in PUF function output errors. How to improve the fault tolerance rate is an urgent problem to be solved.

## 2. Preliminaries

**2.1. Chinese Remainder Theorem [36].** It is assumed that there are  $n$  prime positive integers  $p_1, p_2, \dots, p_n$ . Let  $P$  be the product of  $n$  prime positive integers as  $P = \prod_{i=1}^n p_i$  and  $P_i = P/p_i$ , where  $i = 1, 2, \dots, n$ . Let  $P_i^{-1}$  be the modular multiplicative inverse of  $P_i \pmod{p_i}$  and satisfy  $P_i P_i^{-1} \equiv 1 \pmod{p_i}$ . Then, let  $a_i, i = 1, 2, \dots, n$ , be any  $n$  positive integers. Equation (1) has a unique general solution mod  $P$ .

$$\begin{aligned} X &\equiv a_1 \pmod{p_1} \\ X &\equiv a_2 \pmod{p_2} \\ &\vdots \\ X &\equiv a_n \pmod{p_n} \end{aligned} \quad (1)$$

The general solution of Equation (1) is calculated in Equation (2).

$$\begin{aligned} X &= a_1 P_1^{-1} P_1 + a_2 P_2^{-1} P_2 + \dots + \\ &\quad a_n P_n^{-1} P_n \pmod{P}, \\ &= \sum_{i=1}^n a_i P_i^{-1} P_i \pmod{P}, \\ &= a_1 + a_2 + \dots + a_n \pmod{P}. \end{aligned} \quad (2)$$

**2.2. Physical Unclonable Function [28].** PUF which is based on complex physical system is a function  $F: C \rightarrow R$  ( $C: \{0, 1\}^{\lambda_1}, R: \{0, 1\}^{\lambda_2}$ ). The challenges and their corresponding responses are called challenge-response pairs. PUF has the following properties:

- (1) *Unclonable.* For all  $c \in C$ , there is no function  $F'$  satisfying  $F'(c) = F(c)$ . The probability of duplicating function  $F$  with a cloned function  $F'$  in probabilistic polynomial time (PPT) is negligible
- (2) *Computable.* It is feasible to compute  $r_i = F(c_i)$  in probabilistic polynomial time for all  $c_i \in C$
- (3) *Unpredictable.* For all  $c \in C$ , the probability of the adversary  $\mathcal{A}$  correctly guessing response  $r$  of the function  $F$  corresponding to challenge  $c$  in probabilistic polynomial time is negligible. The output of the function  $F$  is a random string uniformly chosen from  $\{0, 1\}^{\lambda_1}$
- (4) *Tamper-Proofing.* For all  $c, c' \in C$ , even the Hamming distance between  $c$  and  $c'$  is equal to  $t$  ( $t$  is sufficiently small) or less; the probability of outputting the similar results is negligible. Therefore, PUF is able to resist tampering attacks

**2.3. Fuzzy Extractor [5].** The fuzzy extractor takes a low-entropy value containing noise as inputs and outputs the same uniform random value as long as inputs values are close. The fuzzy extractor is utilized to extract the user's biometric information and the smart device's information. It is assumed that fuzzy extractor is composed of two algorithms defined in a tuple  $\langle M, l, t \rangle$ .

*Gen()*: it is a probabilistic algorithm. The user takes his/her biometrics  $BIO_i$  from the metric space  $M$  as  $Gen(BIO_i) = (\sigma_i, \tau_i)$ , and the algorithm outputs the biometric key  $\sigma_i \in \{0, 1\}^l$  and the public parameter  $\tau_i$ .

*Rep()*: it is a deterministic algorithm. Rep takes the biometrics  $BIO_i' \in M$ , reproduction parameter  $\tau_i$ , and  $t$  as the input ( $t$  is the fault tolerance value and sufficiently small). The algorithm Rep can reproduce the biometric key  $\sigma_i$  as  $Rep(BIO_i', \tau_i) = \sigma_i$ , where the Hamming distance between twice inputs is  $t$  or less.

## 3. System Model and Definitions

**3.1. System Model.** The authentication protocol in the smart home consists of the user  $U_i$ , home gateway (HG), smart devices  $SD_j$ , and registration center (RC). All the entities are defined as shown in Figure 1.

- (i) *RC.* RC is usually considered as a trusted registration center. It mainly has two functions including registering the user, HG, and smart devices and generating parameters for smart devices securely
- (ii) *HG.* It is a trusted entity and cannot be compromised by the adversary  $\mathcal{A}$ . It acts as the communication medium between the user and smart devices in the smart home and is responsible for reconstructing secrets for smart devices during the authentication phase
- (iii)  *$U_i$ .* The user  $U_i$  utilizes a smartphone or other smart devices which are referred to as user equipment  $UE_i$ . The user equipment has capability to extract  $U_i$ 's biometrics and verify the authenticity of  $U_i$ 's identity.  $U_i$  can access smart devices after registering at the RC
- (iv)  *$SD_j$ .* Smart devices can execute the commands and collect all kinds of information in the smart home. It is assumed that  $\mathcal{A}$  may attain authentication credentials stored in the smart devices through side-channel attack [21]. PUF technique can be utilized to identify the smart device due to the inherent physical characteristic. All the smart devices have the PUF module which protects them from device capturing attack. Therefore, each smart device cannot be forged physically by the adversary

**3.2. Threat Model.** It is assumed that the adversary  $\mathcal{A}$  in our protocol has same capabilities as the adversary in Dolev-Yao (DY) threat model [37–39]. The capabilities of  $\mathcal{A}$  in our protocol are enumerated as follows:

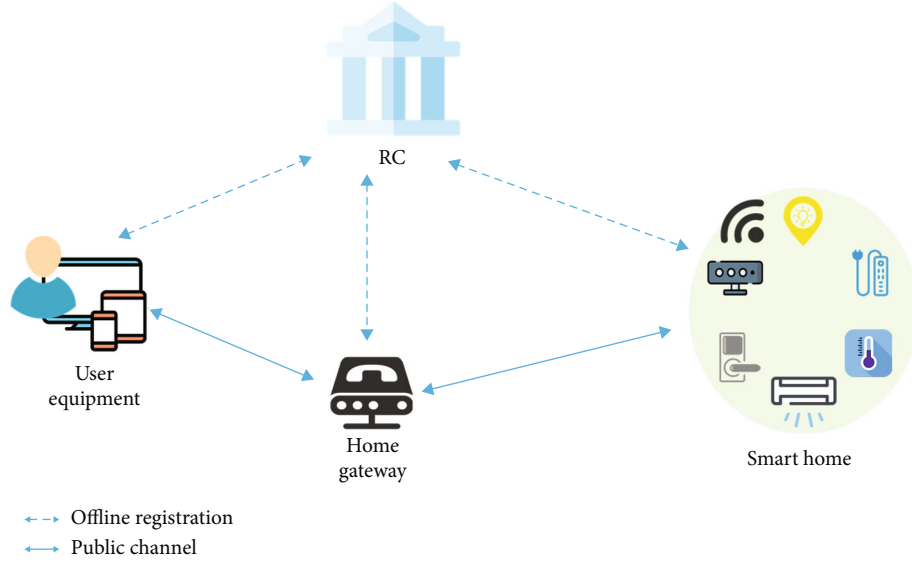


FIGURE 1: System model.

- (i)  $\mathcal{A}$  can eavesdrop, intercept, modify, inject, and delete all the messages transmitted via the public network
- (ii)  $\mathcal{A}$  can store or resend all the messages which are intercepted or forged
- (iii)  $\mathcal{A}$  can impersonate as the legitimate user or the smart device to participate in the authentication process during the execution of the protocol
- (iv)  $\mathcal{A}$  can obtain the credentials stored in the user equipment and launch various types of attacks on the protocol. However, the group session key cannot be compromised to the adversary during the execution of the protocol

In addition, the adversary  $\mathcal{A}$  also has partial abilities in CK-adversary model proposed by Canetti and Krawczyk [40, 41]. Under the CK-adversary model, the reveal of ephemeral state information or other sensitive information has no influence on the security of sessions and long-term secrets. It is necessary to be guaranteed that the security of other sessions cannot be broken even though ephemeral secrets are compromised.

## 4. Our Proposed Protocol

We firstly introduce an overview of the protocol. A detailed description of the protocol is then presented in this section.

**4.1. Overview of the Protocol.** We propose a PUF-assisted lightweight group authentication and key agreement protocol in the smart home. The proposed protocol mainly includes four types of entities: RC, HG, user equipment, and smart devices.

In our protocol, RC plays the role of registration center. RC is responsible for registering other devices. HG acts as an

intermediate device between the user equipment and smart devices and reconstructs the secret for a group of smart devices. Each user has a smartphone or terminal equipment that can read and verify a user's credential. During the login and authentication phase, the user sends the request to HG, and then, HG forwards the requests to a group of target smart devices. After a series of authentication, smart devices generate corresponding responses and send them to HG; HG encrypts the smart devices' responses and forwards them to the user. The user's shared group session key with a group of legal smart devices is securely established. Besides, the user has abilities to update personal password and biometrics locally. To resist replay attack, we assume that all the entities (i.e., users, HG, smart devices) are synchronized with the clock, and the maximum communication delay is  $\Delta T$ .

The detailed notations and corresponding descriptions are summarized in Table 1.

**4.2. Smart Device Registration Phase.** The smart device registration is executed securely in the section. To prevent device capturing attack launched by the adversary, each smart device generates the physical fingerprint based on the physical unclonable function and fuzzy extractor to protect the credentials stored in its memory.

**4.2.1. SDRP1.** The smart device  $SD_j$ ,  $j = 1, 2, \dots, n$ , utilizes the PUF and fuzzy extractor to extract the information to register itself. The smart device  $SD_j$  firstly selects a random nonce  $c_j$  and compute  $r_j = F(c_j)$ . The digital circuits of the smart devices may be influenced by the changes in the external environment, which results in errors in the output of the PUF function. Therefore, the fuzzy extractor is utilized to reduce errors existing in the physical unclonable function.  $SD_j$  computes  $(R_j, h_j) = \text{Gen}(r_j)$  to generate secret  $R_j$  and sends  $R_j$  to RC securely.

TABLE 1: Notations and descriptions.

Notations	Descriptions
RC	Registration center
$U_i, SD_j$ , and HG	$i^{\text{th}}$ user, $j^{\text{th}}$ smart device, and home gateway
$UE_i$	$i^{\text{th}}$ user equipment
$ID_i, ISD_j$ , and $ID_{\text{HG}}$	$U_i$ 's, $SD_j$ , and HG's identity
$PW_i$	$U_i$ 's password
$BIO_i$	$U_i$ 's biometrics
$\text{Gen}(\cdot), \text{Rep}(\cdot)$	Generation and reproduction algorithm of fuzzy extractor
$\sigma_i, R_j$	$U_i$ 's biometrics key, $SD_j$ 's physical key
$\tau_i, x_i, h_j$	Public parameters
$T_i$	Current timestamp
$\Delta T$	Maximum communication delay
$K_{\text{HG}}$	HG's secret key
$K_i$	Symmetric key between $U_i$ and HG
GSK	Group session key between the user and smart devices
$s$	Secret value utilized for secret sharing
$s_j$	$SD_j$ 's secret share
PUF	Physical unclonable function
$H(\cdot)$	One-way hash function
$\oplus, \parallel$	Concatenation and bit-wise XOR operation, respectively

4.2.2. *SDRP2*. When receiving the registration request from smart device  $SD_j$ ,  $j \in \{1, 2, \dots, n\}$ , RC chooses the identity  $ISD_j$  for each smart device and randomly selects a polynomial  $f(x)$  of degree  $t - 1$ :  $f(x) = a_0 + a_1x + \dots + a_{t-1}x^{t-1} \pmod p$ , such that all the coefficients  $a_j, j \in \{1, 2, \dots, t - 1\}$ , and  $s = f(0)$  are in finite field  $\text{GF}(p)$ . RC computes  $H(s)$  and  $s_j = f(x_j)$  ( $x_j$  is public system information related to the smart device  $SD_j$ ). RC randomly selects a prime positive integer  $p_j, j \in \{1, 2, \dots, n\}$  corresponding to smart device  $SD_j$ . Then, RC computes  $P = \prod_{j=1}^n p_j, P_j = P/p_j, j \in \{1, 2, \dots, n\}$ , and  $\chi = \sum_{j=1}^n P_j P_j^{-1} (P_j P_j^{-1} \equiv 1 \pmod{p_j}, \chi \pmod{p_j} \equiv 1)$ . Finally, RC calculates  $RP_j = R_j \oplus p_j$ ,  $\text{share}_j = R_j \oplus s_j$  and sends  $\langle ISD_j, RP_j, \text{share}_j \rangle$  to corresponding smart device  $SD_j$  securely.

4.3. *User Registration Phase*. The user  $U_i$  must register himself at RC when he wants to access the smart home remotely through HG. As shown in Figure 2, the detailed registration process is executed in the following steps.

4.3.1. *URP1*.  $U_i$  firstly chooses an identity  $ID_i$  and high entropy password  $PW_i$  and imprints personal biometric information  $BIO_i$  using the fuzzy extractor in user equipment  $UE_i$ .  $UE_i$  adopts key generation algorithm  $\text{Gen}(\cdot)$  to generate corresponding biometric key  $\sigma_i$  which acts as an element of three-factor authentication and public parameter  $\tau_i$  as  $\text{Gen}(BIO_i) = (\sigma_i, \tau_i)$ . To protect the  $PW_i$  and  $\sigma_i$ ,  $UE_i$  randomly generates a nonce  $a$  and takes personal credentials  $ID_i, PW_i, \sigma_i$ , and  $a$  as input to compute  $RPW_i = H(ID_i \parallel P$

$W_i \parallel \sigma_i) \oplus a$ . Finally,  $UE_i$  securely sends request  $\langle ID_i, RPW_i \rangle$  to RC.

4.3.2. *URP2*. When getting the request  $\langle ID_i, RPW_i \rangle$  from  $U_i$ , RC firstly generates a 1024-bit long-term secret value  $K_{\text{HG}}$  and calculates  $K_i = H(ID_i \parallel K_{\text{HG}})$ ,  $TPW_i = K_i \oplus RPW_i$ . Then, RC generates the anonymous identity  $TID_i$  corresponding to  $ID_i$  and securely sends the information  $\langle TID_i, TPW_i \rangle$  to  $UE_i$ . Finally, RC deletes the information  $RPW_i$  and  $TPW_i$  from its database.

4.3.3. *URP3*. Upon receiving the response  $\langle TID_i, TPW_i \rangle$  from RC,  $UE_i$  computes  $A_i = H(PW_i \parallel \sigma_i \parallel a)$ ,  $B_i = H(ID_i \parallel \sigma_i) \oplus a$ ,  $rPW_i = TPW_i \oplus a$ ,  $V_i = H(H(ID_i \parallel \sigma_i) \parallel A_i) \pmod{\Omega}$ .  $\Omega$  is a medium integer that defines the ability to withstand online guessing attack using "fuzzy-verifier" [42]. Then,  $U_i$  stores  $\langle TID_i, rPW_i, B_i, V_i, \tau_i, \text{Gen}(\cdot), \text{Rep}(\cdot), H(\cdot), t \rangle$  in its memory. Finally,  $UE_i$  deletes  $TPW_i, RPW_i, A_i$  from  $UE_i$  so as to prevent user equipment from compromising sensitive information.

4.4. *Home Gateway Registration Phase*. HG chooses an identity  $ID_{\text{HG}}$  and sends the registration request to RC. Upon receiving the request from HG, RC issues a long-term secret key  $K_{\text{HG}}$ , the user identity  $ID_i$ , corresponding temporal identity  $TID_i$ ,  $H(s)$ , and other public parameters  $h_j, x_j, j \in \{1, 2, \dots, n\}$  to HG securely.

4.5. *Login and Authentication Phase*. Figure 3 gives the summary of login and authentication phase which could be divided into seven steps.

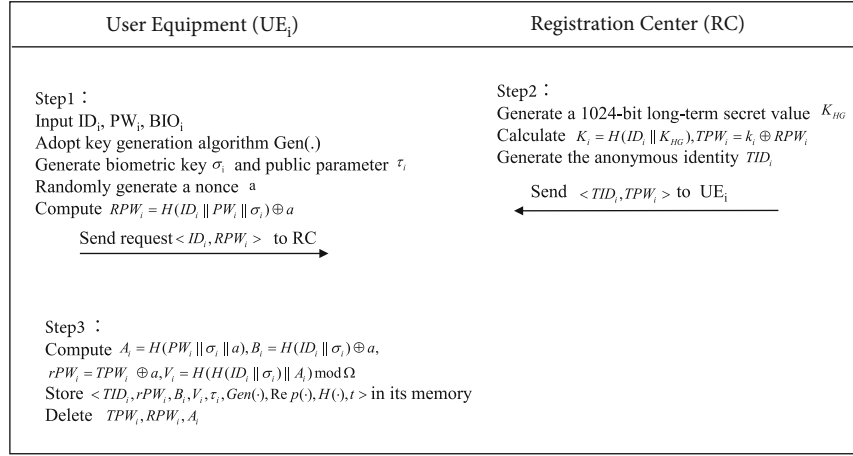


FIGURE 2: Summary of user registration phase.

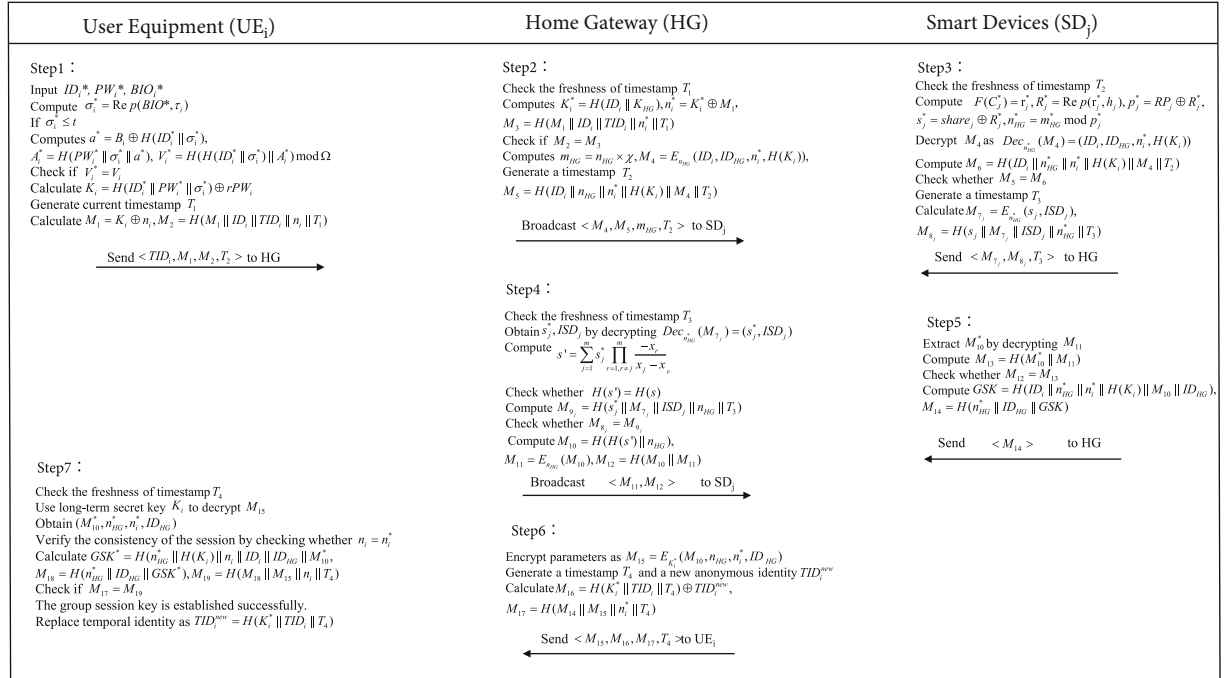


FIGURE 3: Summary of login and authentication phase.

4.5.1. *LAP1*.  $U_i$  firstly inputs  $ID_i^*$  and high entropy password  $PW_i^*$  and imprints personal biometrics  $BIO_i^*$  into  $U_i$ .  $U_i$  computes  $\sigma_i^* = \text{Rep}(BIO_i^*, \tau_i)$  by the reproduction algorithm if the hamming distance between two biometrics is  $t$  or less. Then,  $U_i$  calculates  $a^* = B_i \oplus H(ID_i^* || \sigma_i^*)$ ,  $A_i^* = H(PW_i^* || \sigma_i^* || a^*)$ ,  $V_i^* = H(H(ID_i^* || \sigma_i^*) || A_i^*) \bmod \Omega$ .  $U_i$  verifies the authenticity of the inputs  $ID_i^*$ ,  $PW_i^*$ , and  $BIO_i^*$  by checking whether  $V_i^*$  is equal to the stored  $V_i$ . After verifying the user's identity successfully,  $U_i$  calculates symmetric key  $K_i = H(ID_i^* || PW_i^* || \sigma_i^*) \oplus rPW_i$ .  $U_i$  randomly generates a nonce  $n_i$  and the current timestamp  $T_1$ .  $U_i$  then calculates  $M_1 = K_i \oplus n_i$ ,  $M_2 = H(M_1 || ID_i || TID_i || n_i || T_1)$ .  $U_i$  sends  $\langle TID_i, M_1, M_2, T_1 \rangle$  to HG via an open channel.

4.5.2. *LAP2*. Upon receiving the login request, HG firstly checks the freshness of the timestamp  $T_1$ . If it is true, HG

retrieves  $ID_i$  and  $K_{HG}$ ; computes  $K_i^* = H(ID_i || K_{HG}) = K_i$ ,  $n_i^* = K_i^* \oplus M_1$ , and  $M_3 = H(M_2 || ID_i || TID_i || n_i^* || T_1)$ ; and checks if  $M_2 = M_3$ . If it is invalid, the session is terminated immediately. Then, HG randomly generates a nonce  $n_{HG}$  and a timestamp  $T_2$  and computes  $m_{HG} = n_{HG} \times \chi$ . HG calculates  $M_4 = Enc_{n_{HG}}(ID_i, ID_{HG}, n_i^*, H(K_i))$ ,  $M_5 = H(ID_i || n_{HG} || n_i^* || H(K_i) || M_4 || T_2)$ . Finally, HG broadcasts the message  $\langle M_4, M_5, m_{HG}, T_2 \rangle$  to a group of smart devices via the open channel.

4.5.3. *LAP3*. Upon receiving the message,  $SD_j$  firstly checks the freshness of the message by timestamp  $T_2$ . If it is valid,  $SD_j$  calculates  $F(C_j^*) = r_j^*$ ,  $R_j^* = \text{Rep}(r_j^*, h_j)$ ,  $p_j^* = RP_j \oplus R_j^*$ ,  $s_j^* = \text{share}_j \oplus R_j^*$ ,  $n_{HG}^* = m_{HG} \bmod p_j^*$  ( $\chi \bmod p_j^* \equiv 1$ ,  $n_{HG}^*$  is called as a shared key of a group of legitimate smart devices).

Then,  $SD_j$  decrypts  $M_4$  as  $\text{Dec}_{n_{HG}^*}(M_4) = (ID_i, ID_{HG}, n_i^*, H(K_i))$  using shared group key  $n_{HG}^*$  and computes  $M_6 = H(ID_i \| n_{HG}^* \| n_i^* \| H(K_i) \| M_4 \| T_2)$ . Then,  $SD_j$  checks whether  $M_5 = M_6$ . If it is invalid,  $SD_j$  terminates the session immediately. Otherwise,  $SD_j$  generates a timestamp  $T_3$  and calculates  $M_7 = E_{n_{HG}^*}(s_j, \text{ISD}_j)$ ,  $M_8 = H(s_j \| M_7 \| \text{ISD}_j \| n_{HG}^* \| T_3)$ . Finally,  $SD_j$  sends message  $\langle M_7, M_8, T_3 \rangle$  to HG.

**4.5.4. LAP4.** After receiving  $\langle M_7, M_8, T_3 \rangle$  from smart devices  $SD_j, j \in \{1, 2, \dots, m\}$ . HG checks the freshness of timestamp  $T_3$ . If it is valid, HG can obtain  $s_j^*, \text{ISD}_j$  by decrypting  $\text{Dec}_{n_{HG}^*}(M_7) = (s_j^*, \text{ISD}_j)$  and compute  $s' = \sum_{j=1}^m s_j^* \prod_{r=1, r \neq j}^m (-x_r / (x_j - x_r))$ . HG also checks whether  $H(s') = H(s)$ . If it is true, continues the session. Otherwise, HG computes  $M_9 = H(s_j^* \| M_7 \| \text{ISD}_j \| n_{HG}^* \| T_3)$  and verifies the authenticity of corresponding  $SD_j$  by checking whether  $M_{8_j} = M_9_j$ . If it matches, the message is from valid  $SD_j$ . Otherwise, HG marks  $SD_j$  as invalid smart devices and terminates the session. Then, HG computes  $M_{10} = H(H(s') \| n_{HG}^*)$ ,  $M_{11} = E_{n_{HG}^*}(M_{10})$ ,  $M_{12} = H(M_{10} \| M_{11})$ . Finally, HG sends  $\langle M_{11}, M_{12} \rangle$  to all legitimate smart devices in the group.

**4.5.5. LAP5.** Upon receiving the message  $\langle M_{11}, M_{12} \rangle$ , each smart device  $SD_j$  firstly extracts  $M_{10}^*$  by decrypting the  $M_{11}$  using shared group key  $n_{HG}^*$ , computes  $M_{13} = H(M_{10}^* \| M_{12})$ , and checks whether  $M_{12} = M_{13}$ . If it is valid, each  $SD_j$  computes  $\text{GSK} = H(n_{HG}^* \| H(K_i) \| n_i^* \| ID_i \| ID_{HG} \| M_{10})$ ,  $M_{14} = H(n_{HG}^* \| ID_{HG} \| \text{GSK})$ . Finally, each  $SD_j$  sends the message  $\langle M_{14} \rangle$  to HG.

**4.5.6. LAP6.** HG encrypts parameters as  $M_{15} = E_{K_i^*}(M_{10}, n_{HG}, n_i^*, ID_{HG})$  and generates a timestamp  $T_4$ , a new anonymous identity  $\text{TID}_i^{\text{new}}$ . HG calculates  $M_{16} = H(K_i^* \| \text{TID}_i \| T_4) \oplus \text{TID}_i^{\text{new}}$ ,  $M_{17} = H(M_{14} \| M_{15} \| n_i^* \| T_4)$ . Finally, HG sends the message  $\langle M_{15}, M_{16}, M_{17}, T_4 \rangle$  to  $UE_i$ .

**4.5.7. LAP7.**  $UE_i$  firstly checks the freshness of timestamp  $T_4$  when receiving the message  $\langle M_{15}, M_{16}, M_{17}, T_4 \rangle$ .  $UE_i$  then utilizes long-term secret key  $K_i$  to decrypt  $M_{15}$  and obtains  $(M_{10}^*, n_{HG}^*, n_i^*, ID_{HG})$ .  $UE_i$  verifies the consistency of the session by checking whether  $n_i = n_i^*$ . If it matches,  $U_i$  calculates  $\text{GSK}^* = H(n_{HG}^* \| H(K_i) \| n_i \| ID_i \| ID_{HG} \| M_{10}^*)$ ,  $M_{18} = H(n_{HG}^* \| ID_{HG} \| \text{GSK}^*)$ ,  $M_{19} = H(M_{18} \| M_{15} \| n_i \| T_4)$ .  $UE_i$  checks if  $M_{17} = M_{19}$ . If it matches, the group session key is established successfully. Finally,  $UE_i$  replaces temporal identity as  $\text{TI}_i^{\text{new}} = H(K_i^* \| \text{TID}_i \| T_4) \oplus M_{16}$ .

**4.6. Biometrics and Password Update Phase.** In this section,  $U_i$  can update the password and biometrics in the following steps.

**4.6.1. BPUP1.**  $U_i$  provides personal credentials  $ID_i, \text{PW}_i^{\text{old}}$ , and  $\text{BIO}_i^{\text{old}}$  to  $UE_i$ .  $UE_i$  computes biometrics key  $\sigma_i^{\text{old}}$  as  $\text{Gen}(\text{BIO}_i^{\text{old}}) = (\sigma_i^{\text{old}}, \tau_i^{\text{old}})$  and calculates  $D_i^{\text{old}} = H(ID_i \| \sigma_i^{\text{old}})$ ,  $a^* = B_i \oplus D_i^{\text{old}}$ ,  $A_i^{\text{old}} = H(\text{PW}_i^{\text{old}} \| \sigma_i^{\text{old}} \| a^*)$ , and  $V_i^{\text{old}} = H(D_i^{\text{old}}$

$\| A_i^{\text{old}}) \bmod \Omega$ .  $UE_i$  validates the authenticity of  $U_i$  by checking whether  $V_i^{\text{old}} = V_i$ . If it matches, the user  $U_i$  can update personal password and biometrics. Otherwise,  $UE_i$  terminates the update phase.

**4.6.2. BPUP2.**  $U_i$  enters new password  $\text{PW}_i^{\text{new}}$  and imprints biometrics  $\text{BIO}_i^{\text{new}}$  into the user equipment  $UE_i$ .  $UE_i$  computes  $\sigma_i^{\text{new}}$  as  $\text{Gen}(\text{BIO}_i^{\text{new}}) = (\sigma_i^{\text{new}}, \tau_i^{\text{new}})$  and calculates  $D_i^{\text{new}} = H(ID_i \| \sigma_i^{\text{new}})$ ,  $B_i^{\text{new}} = B_i \oplus D_i^{\text{old}} \oplus D_i^{\text{new}}$ ,  $A_i^{\text{new}} = H(\text{PW}_i^{\text{new}} \| \sigma_i^{\text{new}} \| a^*)$ ,  $r\text{PW}_i^{\text{new}} = r\text{PW}_i \oplus H(ID_i \| \text{PW}_i^{\text{old}} \| \sigma_i^{\text{old}}) \oplus H(ID_i \| \text{PW}_i^{\text{new}} \| \sigma_i^{\text{new}})$ , and  $V_i^{\text{new}} = H(D_i^{\text{new}} \| A_i^{\text{new}}) \bmod \Omega$ . Finally,  $UE_i$  replaces  $B_i, V_i, r\text{PW}_i$ , and  $\tau_i^{\text{old}}$  with  $B_i^{\text{new}}, V_i^{\text{new}}, r\text{PW}_i^{\text{new}}$ , and  $\tau_i^{\text{new}}$  without the help of RC, respectively.

**4.7. Dynamic Smart Devices Joining and Revoking Phase.** Some new smart devices may be added to the smart home after the initial deployment or some deployed smart devices may leave the smart home for some reasons. Therefore, to revoke the defunct device or add the new device into the smart home, it is necessary to update the status of smart devices in real-time. The detailed joining and leaving process is executed in the following steps.

**4.7.1. Joining.** When joining the smart home, a new smart device  $SD_j^{\text{new}}$  must firstly register itself as RC.  $SD_j^{\text{new}}$  randomly chooses a challenge value  $c_j^{\text{new}}$  and generates its physical fingerprint  $R_j^{\text{new}}$  based on PUF and fuzzy extractor technique. Then, a new smart device sends  $R_j^{\text{new}}$  to RC securely. RC generates a unique identity  $\text{ISD}_j^{\text{new}}$  and legitimate share  $(s_j^{\text{new}}, p_j^{\text{new}})$  and computes  $\text{Var}_j^{\text{new}}$ . Then, RC adds  $\text{Var}_j^{\text{new}} = P_j P_j^{-1}$  to  $\chi$  as  $\chi^{\text{new}} = \chi + \text{Var}_j^{\text{new}}$ . During the execution of authentication and key agreement phase, only the legitimate smart devices can calculate secret  $n_{HG}^{\text{new}}$  as  $m_{HG}^{\text{new}} \bmod s_j^{\text{new}} = n_{HG}^{\text{new}}$  ( $m_{HG}^{\text{new}} = n_{HG}^{\text{new}} \times \chi^{\text{new}} \bmod p_j^{\text{new}} \equiv 1, n_{HG}^{\text{new}} < p_j^{\text{new}}$ ). Finally, the new smart devices can be accessed by user  $U_i$ .

**4.7.2. Revoking.** To protect the session security, HG should update the status of smart devices. A smart device that wants to leave the group or is marked as an illegal device will be revoked by HG. The HG subtracts corresponding  $\text{Var}_j^{\text{revoking}}$  from  $\chi$  as  $\chi^{\text{new}} = \chi - \text{Var}_j^{\text{revoking}}$ . The HG generates a new temporal secret and broadcasts it to a group of smart devices. The revoked smart device will fail to compute secret and decrypt the message due to the update of  $\chi^{\text{new}}$ .

## 5. Security Analysis

The widespread Real-or-Random (ROR) model proposed by Abdalla et al. [13] is adopted to establish our security model in this section.

### 5.1. Formal Security Analysis

- (1) *Participants.* Let  $\prod_{U_i}^u, \prod_{SD_j}^v$ , and  $\prod_{HG}^t$  represent instances  $u, v$ , and  $t$  of participant  $U_i, SD_j$ , and HG, respectively

- (2) *Partnering*. If the following conditions are satisfied, the instances  $\prod_{U_i}^u$  and  $\prod_{SD_j}^v$  are said to be partners [37].
- (i) Both instances  $\prod_{U_i}^u$  and  $\prod_{SD_j}^v$  are accepted
  - (ii) Both instances  $\prod_{U_i}^u$  and  $\prod_{SD_j}^v$  authenticate each other
  - (iii) The instance  $\prod_{U_i}^u$  and the instance  $\prod_{SD_j}^v$  are only partners each other
- (3) *Freshness*. The instance  $\prod_{U_i}^u$  or  $\prod_{SD_j}^v$  is fresh if the session key SK is not compromised to  $\mathcal{A}$
- (4) *Adversary*.  $\mathcal{A}$  has all the capabilities as the adversary in Dolev-Yao (DY) threat model [37–39] and also has some capabilities defined in CK-adversary model [40, 41]. Moreover,  $\mathcal{A}$  can make queries as Execute  $(\prod_u, \prod_v)$ , Reveal  $(\prod)$ , Send  $(\prod, m)$ , CorruptUserEquipment  $(\prod_{U_i}^t)$ , CorruptSmartDevice  $(\prod_{SD_j}^t)$ , and Test  $(\prod)$  to challenger to obtain the sensitive information. These queries are utilized to construct a series of games. After games,  $\mathcal{A}$  guesses a bit  $b'$  and wins the game only if  $b' = b$ . Succ represents that  $\mathcal{A}$  wins the game. The advantage of  $\mathcal{A}$  in breaking the IND-CPA of our protocol  $\mathcal{P}$  in probabilistic polynomial time is  $\text{Adv}_{\mathcal{P}, \mathcal{A}}^{\text{IND-CCA}}(\mathcal{K}) = |2 \cdot \Pr [\text{Succ}] - 1|$ . The proposed protocol  $\mathcal{P}$  is secure under the ROR model when  $\text{Adv}_{\mathcal{P}, \mathcal{A}}^{\text{IND-CPA}}(\mathcal{K})$  is negligible

**Theorem 1.** Let  $\mathcal{A}$  be the adversary running in the polynomial time  $t$  against our authentication protocol  $\mathcal{P}$  in the random oracle. Let Dic,  $q_h$ ,  $q_{\text{send}}$ ,  $q_e$ ,  $|\text{Hash}|$ ,  $|\text{Dic}|$ ,  $m$ , and  $l'$  represent the a uniformly distributed password dictionary, the number of Hash oracles, the number of Send oracle, the number of Execute oracles, the space of hash function, the size of Dic, the bit length of biometrics key  $\sigma_i$ , and the bit length of the random nonce, respectively. The advantage of  $\mathcal{A}$  in breaking protocol  $\mathcal{P}$  in probabilistic polynomial time is defined as follows:

$$\text{Adv}_{\mathcal{P}, \mathcal{A}}^{\text{AKA}}(\mathcal{K}) \leq \frac{q_h^2}{|\text{Hash}|} + \frac{(q_{\text{send}} + q_e)^2}{2^{l'}} + 2 \max \left( \frac{q_{\text{send}}}{2^m}, C' \cdot q_{\text{send}}^{s'} \right) + \frac{2}{q} \cdot \text{Adv}_{\mathcal{P}, \mathcal{A}}^{\text{IND-CPA}}(\mathcal{K}). \quad (3)$$

*Proof.* The games  $\text{Game}_i$ , where  $i = [0, 4]$  is defined in this section. Let  $\text{Succ}_i$  represent the event that  $\mathcal{A}$  succeeds in guessing  $b$  in the  $\text{Game}_i$ .

$\text{Game}_0$ : the game  $\text{Game}_0$  simulates the real attack in our protocol by  $\mathcal{A}$  in ROR sense. At the beginning of  $\text{Game}_0$ ,  $\mathcal{A}$  guesses  $b$ . By definition, it follows

$$\text{Adv}_{\mathcal{P}, \mathcal{A}}^{\text{AKA}}(\mathcal{K}) = |2 \Pr [\text{Succ}_0] - 1|. \quad (4)$$

$\text{Game}_1$ : the game  $\text{Game}_1$  simulates the adversary's eavesdropping attack by asking Execute  $(\prod, \prod)$  oracle. At the end of the game,  $\mathcal{A}$  queries Test oracle and then distinguishes whether the output of Test oracle is either a real session key SK or a random string in the same domain. The group session key is calculated as  $\text{GSK} = H(n_{\text{HG}} \| H(K_i) \| n_i \| \text{ID}_i \| \text{ID}_{\text{HG}} \| H(H(s) \| n_{\text{HG}}))$  in our protocol. To calculate the GSK,  $\mathcal{A}$  has to obtain  $H(K_i)$  and  $H(H(s) \| n_{\text{HG}})$ . Additionally,  $\text{ID}_i$ ,  $\text{ID}_{\text{HG}}$ ,  $n_i$ , and  $n_{\text{HG}}$  are not compromised to  $\mathcal{A}$ . Therefore, the probability of winning  $\text{Game}_1$  for  $\mathcal{A}$  is not increased by launching eavesdropping attacks. It is clear that

$$\Pr [\text{Succ}_0] = \Pr [\text{Succ}_1]. \quad (5)$$

$\text{Game}_2$ : there exists some differences between  $\text{Game}_2$  and  $\text{Game}_1$ ; the simulations of Send and Hash oracles are added to the  $\text{Game}_2$ . The game simulates an active attack in which  $\mathcal{A}$  tries to fool the participant into accepting the forged messages.  $\mathcal{A}$  is able to query Hash oracle many times to find collisions. Since all the exchanged messages are associated with participant's identity, random nonce, and timestamps, the probability of finding the collision of secret key for symmetric cryptography is  $q_h^2/2 \cdot |\text{Hash}|$  according to the birthday paradox. Besides, the probability of finding the collision of random nonce is defined as  $(q_{\text{send}} + q_e)^2/2^{l'+1}$ . It is clear that

$$|\Pr [\text{Succ}_1] - \Pr [\text{Succ}_2]| \leq \frac{q_h^2}{2 \cdot |\text{Hash}|} + \frac{(q_{\text{send}} + q_e)^2}{2^{l'+1}}. \quad (6)$$

$\text{Game}_3$ : by adding the simulation of querying the CorruptSmartPhone oracle and smartphone lost attack, the  $\text{Game}_2$  is transformed into  $\text{Game}_3$ .  $\mathcal{A}$  may obtain password  $\text{PW}_i$  and the biometrics key  $\sigma_i$  using online, offline dictionary attack, and physical device attack, respectively. The fuzzy extractor is utilized to extract the  $b$  bits of biometric information, and the probability of guessing the  $\sigma_i \in \{0, 1\}^m$  for  $\mathcal{A}$  is approximately  $1/2^m$ . Additionally, it is supposed that the number of password inputs is strictly limited. The user-chosen passwords tend to be low entropy and are far different distribution from uniform distribution. The size of the password space is limited in practical, and users usually only use a part of the password space. The probability of guessing the password is defined as  $C' \cdot q_{\text{send}}^{s'}$  [43];  $C'$  and  $s'$  are the parameters of the Zipf model. Therefore, it is clear that

$$|\Pr [\text{Succ}_2] - \Pr [\text{Succ}_3]| \leq \max \left( \frac{q_{\text{send}}}{2^m}, C' \cdot q_{\text{send}}^{s'} \right). \quad (7)$$

$\text{Game}_4$ : this game adds the simulation of CorruptSmartDevice oracle compared to  $\text{Game}_3$ .  $\mathcal{A}$  can physically capture the smart devices and obtain the information prestored into the memory of smart device in the registration phase. However, this information is encrypted by the physical fingerprint  $R_j$  based on PUF and fuzzy extractor



technique. It is hard to obtain the secret share  $s_j$  and forge the device even if  $\mathcal{A}$  grabs the device. Let  $\mathcal{A}$  can eavesdrop all the exchanged messages.  $\mathcal{A}$  tries to obtain the sensitive information  $\{ID_i, ID_{HG}, n_i, M_{10}, H(K_i)\}$  by decrypting the message  $M_4$ . Due to the Chinese Remainder Theorem, any illegitimate participant is unable to obtain the temporary group key  $n_{HG}$  and  $H(K_i)$  without the secret share  $s_j$ . Even if  $\mathcal{A}$  wants to reconstruct secret, it is hard for  $\mathcal{A}$  to capture at least  $t$  legal smart devices. The probability of forging the appropriate pair of values is  $1/q$ . Additionally, it is difficult for  $\mathcal{A}$  to decrypt the  $M_{15}$  as  $\mathcal{A}$  is unknown to  $K_i$ .  $\mathcal{A}$  can not compute  $GSK = H(n_{HG} \| H(K_i) \| n_i \| ID_i \| ID_{HG} \| H(H(s) \| n_{HG}))$  due to the lacking of  $H(H(s) \| n_{HG})$  and  $H(K_i)$ . The proposed protocol is IND – CPA secure. It is concluded that

$$|\Pr [\text{Succ}_3] - \Pr [\text{Succ}_4]| \leq \frac{1}{q} \cdot \text{Adv}_{\mathcal{P}, \mathcal{A}}^{\text{IND-CPA}}(\mathcal{K}). \quad (8)$$

All the oracles have been simulated in the game.  $\mathcal{A}$  guesses  $b$  after querying Test oracle. It is clear that  $\Pr [\text{Succ}_4] = 1/2$ .

Therefore, from formulas (4) to (8), we have

$$\begin{aligned} \text{Adv}_{\mathcal{P}, \mathcal{A}}^{\text{AKA}}(\mathcal{K}) &= 2 \cdot \left| \Pr [\text{Succ}_0] - \frac{1}{2} \right| = 2 \cdot |\Pr [\text{Succ}_1] - \Pr [\text{Succ}_4]| \\ &\leq 2 \cdot (|\Pr [\text{Succ}_1] - \Pr [\text{Succ}_2]| + |\Pr [\text{Succ}_2] - \Pr [\text{Succ}_4]|) \\ &\leq 2 \cdot (|\Pr [\text{Succ}_1] - \Pr [\text{Succ}_2]| + |\Pr [\text{Succ}_2] - \Pr [\text{Succ}_3]| + |\Pr [\text{Succ}_3] - \Pr [\text{Succ}_4]|) \\ &\leq 2 \cdot \left( \frac{q_h^2}{2 \cdot |\text{Hash}|} + \frac{(q_{\text{send}} + q_e)^2}{2^{f+1}} + \max \left( \frac{q_{\text{send}}}{2^m}, C' \cdot q_{\text{send}}' \right) + \frac{1}{q} \cdot \text{Adv}_{\mathcal{P}, \mathcal{A}}^{\text{IND-CPA}}(\mathcal{K}) \right) \\ &\leq \frac{q_h^2}{|\text{Hash}|} + \frac{(q_{\text{send}} + q_e)^2}{2^f} + 2 \max \left( \frac{q_{\text{send}}}{2^m}, C' \cdot q_{\text{send}}' \right) \\ &\quad + \frac{2}{q} \cdot \text{Adv}_{\mathcal{P}, \mathcal{A}}^{\text{IND-CPA}}(\mathcal{K}). \end{aligned} \quad (9)$$

□

## 5.2. Other Discussions on Security Features

**5.2.1. Untraceability and User Anonymity.** It is assumed that  $\mathcal{A}$  has capability of intercepting all the messages during the execution of the authentication phase over the public channel. The user's identity  $ID_i$  is protected by hash function  $H(\cdot)$  and symmetric cryptography. It is computationally infeasible for  $\mathcal{A}$  to attain identity without secret parameters  $n_{HG}, n_i, V_i, \sigma_i$ . Therefore, our protocol guarantees the feature of user anonymity. Moreover, the transmitted message generally involves the current timestamp and random nonce, and  $U_i$  temporary identity  $TID_i$  is updated when the session is completed successfully. Therefore, it is also computationally infeasible for  $\mathcal{A}$  to track the user's activity in each session. In conclusion, the untraceability and user anonymity are both guaranteed in our protocol.

**5.2.2. Replay Attack.** It is assumed that  $\mathcal{A}$  is capable of intercepting all the messages between the user, HG, and smart devices. The transmitted messages usually involve random nonces and timestamps. Even if  $\mathcal{A}$  intercepts the messages and replays these messages shortly after, they can not pass the verification of timestamps due to maximum communication delay  $\Delta T$ . Thus, our protocol can resist replay attack.

**5.2.3. Smart Device Impersonation Attack.** It is supposed that  $\mathcal{A}$  intercepts the transmitted message during the execution of the protocol.  $\mathcal{A}$  needs to generate valid information. However,  $\mathcal{A}$  does not know the sensitive parameters to obtain the authentication parameters. Furthermore, the smart device is protected by PUF, which cannot be forged on hardware. It is computationally infeasible to impersonate the smart device in probabilistic polynomial time. Therefore, our protocol can withstand smart device impersonation attack.

**5.2.4. HG Impersonation Attack.** It is supposed that  $\mathcal{A}$  intercepts the message during the execution of the protocol and tries to generate other messages to impersonate HG. However, without the knowledge of the secret parameters  $\chi, n_i, ID_i, K_{HG}$ , it is computationally infeasible to impersonate HG in probabilistic polynomial time. Thus, our protocol can withstand HG impersonation attack.

**5.2.5. Smartphone Lost Attack.** Supposed that the  $U_i$ 's smartphone is lost or stolen by  $\mathcal{A}$ . By the threat model,  $\mathcal{A}$  is capable of extracting all the information  $\{TID_i, rPW_i, B_i, V_i, \tau_i, \text{Gen}(\cdot), \text{Rep}(\cdot), H(\cdot), t\}$  stored in the memory of  $UE_i$  using the power analysis attack [44]. In order to retrieve  $ID_i, PW_i$  from the extracted information  $\mathcal{A}$  needs to attain the secrets  $K_i, \sigma_i, n_i$ . The possibility of guessing the user's biometrics key  $\sigma_i$  as well as  $n_i, K_i$  is negligible. The adversary  $\mathcal{A}$  may launch the password guessing attack. The password guessing attack is mainly divided into online and offline password guessing attack [45]. The online password guessing attack can be effectively prevented by limiting the number of illegal requests from users. In our paper, the "fuzzy verifier" is utilized to guarantee the security under offline password guessing attack. The password verifier  $V_i$  is computed  $V_i = H(H(ID_i \| \sigma_i) \| H(PW_i \| \sigma_i) \| a) \bmod \Omega$ . Even if other two authentication factors are compromised, the adversary  $\mathcal{A}$  has to guess  $ID_i, PW_i$ , and  $a$ . Furthermore, it is assumed that  $\mathcal{A}$  has got the  $ID_i^*, PW_i^*$ , and  $a^*$  which satisfying  $V_i = V_i^*$ ; the login request will be rejected due to the "fuzzy verifier." Therefore, our protocol can effectively withstand online and offline guessing attack. The user's identity credentials  $ID_i, PW_i$  are not compromised to  $\mathcal{A}$ . So, our protocol can resist smartphone lost attack.

**5.2.6. Privileged-Insider Attack.** It is assumed that  $\mathcal{A}$  is a privileged-insider user of trusted RC.  $\mathcal{A}$  tries to attain the credentials of the authorized user and all the information from  $UE_i$ .  $\mathcal{A}$  obtains the registration information  $\{ID_i, RPW_i\}$  of  $U_i$  which is sent to RC. Meanwhile,  $\mathcal{A}$  is able to extract all the information  $\{TID_i, rPW_i, B_i, V_i, \tau_i, \text{Gen}(\cdot), \text{Rep}(\cdot), H(\cdot), t\}$  stored in the  $UE_i$ . Without knowing of random nonce  $a$  and biometrics key  $\sigma_i$ , it is computationally infeasible to retrieve  $PW_i$  in probabilistic polynomial time due to  $RPW_i = H(ID_i \| PW_i \| \sigma_i)$ . Thus, our protocol can withstand privileged-insider attack.

**5.2.7. Ephemeral Secret Leakage Attack.** In our protocol, a secure group session key  $GSK^* = H(n_{HG}^* \| H(K_i) \| n_i \| ID_i \| I$

TABLE 2: Security feature comparison.

Feature	[20]	[8]	[9]	[46]	[47]	[48]	Our protocol
User anonymity	√	√	√	√	√	√	√
Untraceability	√	√	√	√	√	√	√
Mutual authentication	√	√	√	√	√	√	√
Perfect forward secrecy	×	×	√	√	√	√	√
Dynamically devices joining	√	√	×	×	√	√	√
Device revocation	√	×	×	×	√	√	√
The number of factors used	Three	Three	Two	N/A	Two	Three	Two
Password/biometrics update	√	√	√	×	×	√	√
Smartphone/smartcard lost attack	√	√	√	N/A	×	√	√
Smart device lost attack	√	√	√	N/A	√	×	√
User impersonation attack	√	√	√	√	√	√	√
Device impersonation attack	√	√	√	N/A	√	√	√
HG impersonation attack	N/A	√	N/A	√	√	√	√
Session key security	√	√	√	√	√	√	√
Replay attack	√	√	√	√	√	√	√
Privileged-insider attack	√	√	√	√	√	√	√
Ephemeral secret leakage attack	N/A	N/A	N/A	×	×	√	√

<sup>1</sup>N/A means not considered. <sup>2</sup>√ means the scheme supports the functionality/security feature. <sup>3</sup>× means the scheme does not support the functionality/security feature.

TABLE 3: Communication cost comparison.

Scheme	Single device cost	$n$ devices cost	The no. of message
Challa et al. [20]	2016	2016 $n$	4
Wazid et al. [8]	2592	2592 $n$	4
Li et al. [9]	2048	2048 $n$	4
Yu and Li [46]	4096	4096 $n$	8
Shuai et al. [47]	2272	2272 $n$	4
Banerjee et al. [48]	2048	2048 $n$	4
Our protocol	3296	1376 + 1920 $n$	6

$D_{HG} \| M_{10}$ ) is established between a user and smart devices during the login and authentication phase.  $M_{10}$  is composed of long-term secret  $H(s)$  and short-term secret  $n_{HG}$ . In particular, the secret  $s$  is computed by secret reconstruction algorithm of secret sharing technology. In addition,  $ID_{HG}$ ,  $ID_i$ ,  $H(K_i)$  are the long-term secrets, and  $n_i$  is a short-term secret. On the one hand, it is assumed that the short-term secrets  $n_{HG}$ ,  $n_i$  are revealed to  $\mathcal{A}$ . However, it is computationally infeasible to compute the GSK due to the lack of long-term secrets. On the other hand, it is assumed that  $\mathcal{A}$  can obtain the long-term secrets. Even though  $\mathcal{A}$  obtains some secret shares  $s_j$  from the smart devices, it is computationally infeasible to construct the secret  $S$  and then calculate the message  $M_{10}$ . The short-term secrets  $n_{HG}$ ,  $n_i$  are ran-

domly generated by the HG and  $U_i$ . It is also hard for  $\mathcal{A}$  to compute GSK without the short-term secrets  $n_{HG}$ ,  $n_i$ . Therefore,  $\mathcal{A}$  cannot compute the current session key unless both all the long-term secrets and short-term secrets are compromised simultaneously. Our protocol can thwart the ephemeral secret leakage attack.

**5.2.8. Perfect Forward Secrecy.** It is supposed that the adversary obtains the secret keys of a user and the smart devices. Furthermore, the adversary intercepts all the messages transmitted among them during the group authentication process. The adversary computes  $GSK = H(n_{HG} \| H(K_i) \| n_i \| ID_i \| ID_{HG} \| M_{10}) = H(n_{HG} \| H(ID_i \| K_{HG}) \| n_i \| ID_i \| ID_{HG} \| H(H(s) \| n_{HG}))$  to get the group session key. However, the adversary cannot obtain the parameters  $n_{HG}$ ,  $K_{HG}$  and reconstruct correctly the secret  $s$  with given shares to compute the group session key. Therefore, the proposed protocol can provide the perfect forward secrecy.

**5.2.9. Session Key Security.** The session key GSK is calculated by both all the authenticated smart devices and the user  $U_i$ . The message  $M_{14}$  contains the session key. Supposed that  $\mathcal{A}$  intercepts the message and tries to forge  $GSK'$  by random nonces  $n_i$ ,  $n_{HG}'$ . However,  $\mathcal{A}$  does not know the parameters  $ID_i$ ,  $H(K_i)$ ,  $M_{10}$ ; it is impossible for  $\mathcal{A}$  to compute GSK due to the collision resistance property of  $H(\cdot)$ . Thus, our protocol guarantees session key security successfully.

## 6. Performance Analysis

We analyze the performance of our protocol from three aspects, including computational cost, communication cost,

TABLE 4: Computational cost comparison.

Protocol	Single device accessing cost (ms)	$n$ devices accessing cost (ms)
Challa et al. [20]	$T_{fe} + 16T_H + 13T_{ecm}$	$(T_{fe} + 16T_H + 13T_{ecm})n$
Wazid et al. [8]	$T_{fe} + 21T_H + 8T_{E/D}$	$(T_{fe} + 21T_H + 8T_{E/D})n$
Li et al. [9]	$T_{fe} + 19T_H + 8T_{E/D} + 3T_{ecm}$	$(T_{fe} + 19T_H + 8T_{E/D} + 3T_{ecm})n$
Yu and Li [46]	$4T_B + 26T_H + 47T_{ecm}$	$(4T_B + 26T_H + 47T_{ecm})n$
Shuai et al. [47]	$16T_H + 8T_{ecm}$	$(16T_H + 8T_{ecm})n$
Banerjee et al. [48]	$T_{fe} + 24T_H$	$(T_{fe} + 24T_H)n$
Our protocol	$T_{puf} + 2T_{fe} + 22T_H + 8T_{E/D}$	$T_{fe} + 9T_H + 4T_{E/D} + (T_{puf} + T_{fe} + 4T_{E/D} + 13T_H)n$

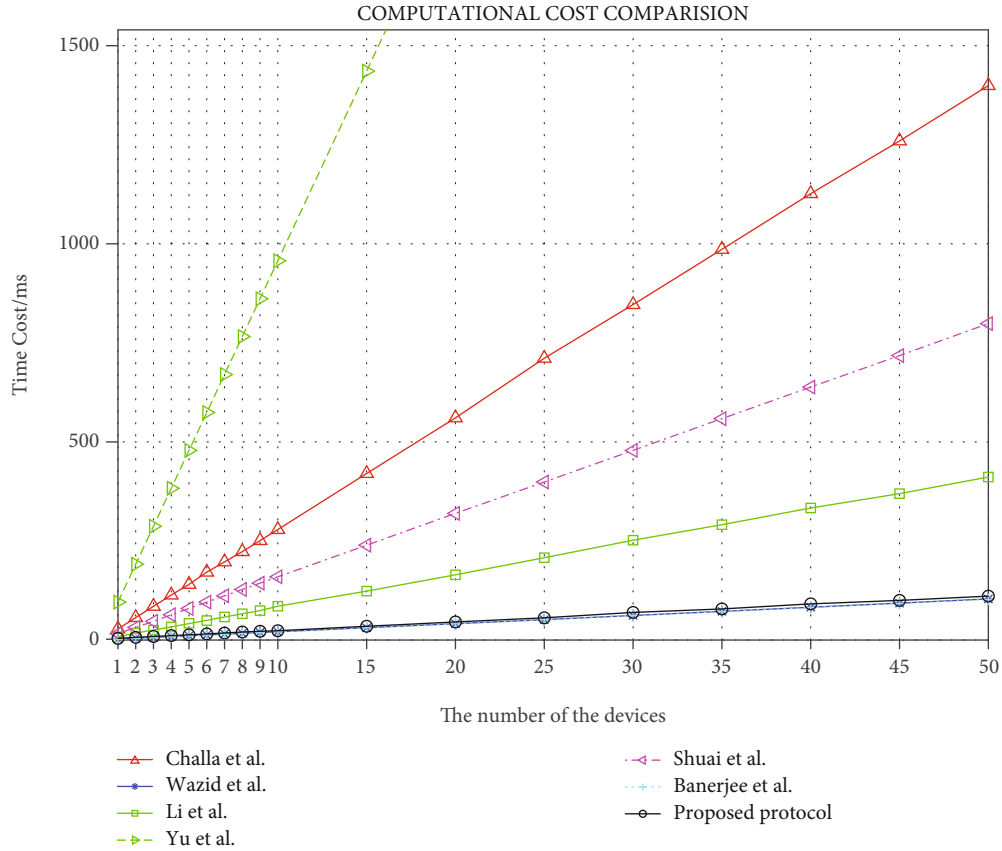


FIGURE 4: Computational cost comparison.

functionality, and security features, respectively. We also compare our protocols with other related protocols in the section.

**6.1. Functionality and Features.** We compare the functionality and security features of our protocol with other related protocols in Table 2. From Table 2, most protocols generally adopt a multifactor authentication mechanism to verify the authenticity of the user. Challa et al. [20] and Li et al. [9]'s protocols are insecure against HG impersonation attack and do not provide perfect forward secrecy. Although most authentication and key agreement protocols for the smart home declare they can resist many known attacks such as

replay attack, privileged-insider attack, and man-in-the-middle attack, most protocols do not support all above features. It is obvious that the proposed protocol still provides more security functionalities and security features than other related protocols [46–48]. Yu and Li [46], Shuai et al. [47], and Banerjee et al. [48] all lack the security protection for the smart devices. The sensitive information stored in the smart devices may be compromised to the adversary while the adversary launch attacks on smart devices. Additionally, Yu and Li [46] and Shuai et al. [47] utilize pairing-based cryptography and ECC-based to implement authentication and establish session key between users and devices, respectively, which are not great for resource-constrained devices.

**6.2. Communication Cost.** We evaluate the communication and computational cost in our authentication protocol compared to other protocols [8, 9, 20, 46–48].

It is defined that the length of identity, random nonces, timestamps, and hash function operation is 128 bits, 128 bits, 32 bits, and 160 bits, respectively. It is also assumed that  $|\lambda_1| = 128$  bits,  $|\lambda_2| = 160$  bits, and AES-128 are adopted for symmetric cryptography, where  $\lambda_1, \lambda_2$  denote the length of input and output of physical unclonable function, respectively. The messages in our protocol include  $\text{msg}_1 = \{\text{TID}_i, M_1, M_2, T_1\}$ ,  $\text{msg}_2 = \{M_4, M_5, m_{\text{HG}}, T_2\}$ ,  $\text{msg}_3 = \{M_{7\text{-SD}_j}, M_{8\text{-SD}_j}, T_3\}$ ,  $\text{msg}_4 = \{M_{11}, M_{12}\}$ ,  $\text{msg}_5 = \{M_{14}\}$ , and  $\text{msg}_6 = \{M_{15}, M_{16}, M_{17}, T_4\}$ ; the corresponding bit length of messages is 480 bits, 864 bits, 576 bits, 320 bits, 160 bits, and 896 bits, respectively. Table 3 summarizes the proposed protocol and other existing authentication protocols in terms of communication cost. The proposed protocol requires second highest communication cost among all the protocols when users launch the access request to single device in the smart home. However, it is obvious that the proposed protocol effectively reduces the communication cost when accessing multiple devices compared to other protocols.

**6.3. Computational Cost.** The proposed protocol is simulated using Pair-Based Cryptography (PBC) library and GNU Multiple Precision Arithmetic (GMP) library. C language is utilized on Ubuntu 16.04 with 2.50 GHz Intel(R) Core(TM) i5-4200M CPU and 8 GB of RAM.

We compare the total execution time with other protocols [8, 9, 20, 46–48] during the login and authentication phase. It is assumed that  $T_B, T_H, T_{E/D}, T_{fe}, T_{xor}, T_{ecm}, T_{mm}, T_{puf}, T_{mac}$ , and  $T_{hmac}$  denote the computational cost required for a bilinear pairing, hash function, a symmetric cryptography using AES-128, a fuzzy extraction operation, a XOR operation, a point multiplication operation using ECC, a modular multiplication operation, a physical unclonable function operation, a message authentication code (MAC) operation, and a hashed MAC operation, respectively. As the computational cost of bit-wise XOR operation is much less than other operations, it is not considered in the evaluation. Besides, it is assumed that  $T_H \approx T_{mac} \approx T_{hmac}$ ,  $T_{fe} \approx T_{ecm}$  in our experiment according to [8]. The above operations are performed one hundred times and take its average value. Based on the experimental results reported in [49], we have the computational cost of  $T_B, T_H, T_{E/D}, T_{fe}, T_{mm}, T_{ecm}$ , and  $T_{puf}$  which is 0.544 ms, 0.0026 ms, 0.00325 ms, 1.989 ms, 0.171 ms, 1.989 ms, and 0.12 ms (ms is the abbreviation of milliseconds), respectively. The computational cost of accessing single and multiple devices for the related protocol and our protocol is described in Table 4. It is clear that the proposed protocol has significantly reduced the computational cost compared to Challa et al. [20] and Shuai et al. [47]. By introducing the Chinese residual theorem and secret sharing, although the copu is performance in the case of single device access, the performance is significantly better in the case of multiple devices access.

Figure 4 shows the comparison of computational cost in the login and authentication phase. Viewed from Figure 4, the  $X$ -axis represents the numbers of smart devices that users access simultaneously. The  $Y$ -axis represents the time cost to establish session key with  $n$  smart devices, simultaneously. It is obvious that the computational cost of Yu and Li [46] is much more than that of other protocols. Compared to protocols of Challa et al. [20], Li et al. [9], and Shuai et al. [47], the protocols of Wazid et al. [8] and Banerjee et al. [48] and our proposed protocol have the similar computational cost when accessing smart devices. Obviously, according to Table 4, the computational complexity of previous schemes increases linearly according to the number of devices. In this scenario, the computation cost is  $T_{fe} + 9T_H + 4T_{E/D} + (T_{puf} + T_{fe} + 4T_{E/D} + 13T_H)n$ . When  $n$  is large, we believe that the constant term can be ignored, so our computation time also increases linearly with the number of devices. However, our protocol effectively supports more functionalities and security features at the cost of slightly increasing the communication and computational cost compared to Wazid et al. [8] and Banerjee et al. [48]'s protocols.

## 7. Conclusion

In this paper, we proposed a PUF-assisted lightweight group authentication and key agreement protocol in the smart home based on secret sharing technique and Chinese Remainder Theorem. The proposed protocol can withstand most of several known attacks, which is proved under the ROR model and other security discussions. Compared with other related protocols, our protocol can effectively reduce the resource cost during the login and authentication phase. In addition, our smart devices protected by the physical unclonable function are secure against smart device lost attack. Our protocol supports dynamic smart device joining and leaving, password, and biometrics update without the involvement of HG. Overall, the performance of our authentication protocol is better than other related protocols only using lightweight operations. Therefore, our protocol is more suitable for resource-constrained smart devices in the smart home. In future work, we will take tools such as AVISPA for further security analysis and verify the performance of the protocol in the smart home.

## Data Availability

The related data used to support the findings of this study are included within the article.

## Disclosure

The paper is extended from the one that is accepted in SPNCE 2020. The previous version can be found at the SPNCE 2020 proceedings.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. 61922045, No. U21A20465, No. 62172292, and No.61877034.

## References

- [1] V. Ricquebourg, D. Menga, D. Durand, B. Marhic, L. Delahoche, and C. Loge, "The smart home concept: our immediate future," in *2006 1ST IEEE International Conference on E-Learning in Industrial Electronics*, pp. 23–28, Hammamet, Tunisia, 2006.
- [2] L. Jiang, D. Liu, and B. Yang, "Smart home research," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, vol. 2, pp. 659–663, Shanghai, China, 2004.
- [3] M. Chiang and T. Zhang, "Fog and IoT: an overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.
- [4] L. Jiang, C. Wang, and J. Shen, "Stereo storage structure assisted one-way anonymous auditing protocol in e-health system," *Journal of Surveillance, Security and Safety*, vol. 1, pp. 61–78, 2020.
- [5] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.
- [6] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ECC-based provable secure authentication protocol with privacy preserving for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3599–3609, 2018.
- [7] X. Ye and J. Huang, "A framework for cloud-based smart home," in *Proceedings of 2011 International Conference on Computer Science and Network Technology*, vol. 2, pp. 894–897, Harbin, 2011.
- [8] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, "Design of secure user authenticated key management protocol for generic IoT networks," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 269–282, 2018.
- [9] X. Li, J. Peng, J. Niu, F. Wu, J. Liao, and K. R. Choo, "A robust and energy efficient authentication protocol for industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1606–1615, 2018.
- [10] Y. Wen and Y. Lao, "Efficient fuzzy extractor implementations for PUF based authentication," in *2017 12th International Conference on Malicious and Unwanted Software (MALWARE)*, pp. 119–125, IEEE Computer Society, Los Alamitos, CA, USA, 2017.
- [11] C. Herder, M. D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: a tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, 2014.
- [12] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, J. J. P. C. Rodrigues, and Y. Park, "Physically secure lightweight anonymous user authentication protocol for Internet of Things using physically unclonable functions," *IEEE Access*, vol. 7, pp. 85627–85644, 2019.
- [13] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," *IEEE Proceedings-Information Security*, vol. 153, no. 1, pp. 27–39, 2006.
- [14] K. Xue, C. Ma, P. Hong, and R. Ding, "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 316–323, 2013.
- [15] D. He, N. Kumar, and N. Chilamkurti, "A secure temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks," in *International Symposium on Wireless and pervasive Computing (ISWPC)* pp. 1–6, Taipei, Taiwan, 2013.
- [16] P. Kumar, A. Braeken, A. Gurtov, J. Iinatti, and P. H. Ha, "Anonymous secure framework in connected smart home environments," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 968–979, 2017.
- [17] M. Turkanovic, B. Brumen, and M. Holbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Networks*, vol. 20, pp. 96–112, 2014.
- [18] S. Kalra and S. K. Sood, "Advanced password based authentication scheme for wireless sensor networks," *Journal of information security and applications*, vol. 20, pp. 37–46, 2015.
- [19] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *Journal of Network and Computer Applications*, vol. 106, pp. 117–123, 2018.
- [20] S. Challa, M. Wazid, A. K. Das et al., "Secure signature-based authenticated key establishment scheme for future IoT applications," *Ieee Access*, vol. 5, pp. 3028–3043, 2017.
- [21] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for Industrial Internet of Things," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, pp. 1133–1146, 2018.
- [22] P. Gope, J. Lee, and T. Q. S. Quek, "Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2831–2843, 2018.
- [23] L. Harn, "Group authentication," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1893–1898, 2013.
- [24] Y. Liu, C. Cheng, J. Cao, and T. Jiang, "An improved authenticated group key transfer protocol based on secret sharing," *IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2335–2336, 2013.
- [25] J. Li, M. Wen, and T. Zhang, "Group-based authentication and key agreement with dynamic policy updating for MTC in LTE-A networks," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 408–417, 2016.
- [26] J. Cui, D. Wu, J. Zhang, Y. Xu, and H. Zhong, "An efficient authentication scheme based on semi-trusted authority in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2972–2986, 2019.
- [27] S. Zhang and J. Lee, "A group signature and authentication scheme for blockchain-based mobile-edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4557–4565, 2020.
- [28] J. R. Wallrabenstein, "Practical and secure IoT device authentication using physical unclonable functions," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 99–106, Vienna, Austria, 2016.
- [29] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 4957–4968, 2019.

- [30] F. L. Tiplea and C. Hristea, "PUF protected variables: a solution to RFID security and privacy under corruption with temporary state disclosure," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 999–1013, 2021.
- [31] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A PUF-based secure communication protocol for IoT," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 3, pp. 1–25, 2017.
- [32] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, no. 8, p. 352, 2018.
- [33] U. Chatterjee, V. Govindan, R. Sadhukhan et al., "Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 3, pp. 424–437, 2019.
- [34] T. Li and Y. Liu, "A double PUF-based RFID authentication protocol," *Journal of Computer Research and Development*, vol. 58, no. 8, pp. 1801–1810, 2021.
- [35] S. Chen, B. Li, Z. Chen, Y. Zhang, C. Wang, and C. Tao, "Novel strong-PUFbased authentication protocols leveraging Shamir's secret sharing," *IEEE Internet of Things Journal*, 2021.
- [36] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "PA-CRT: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 722–735, 2021.
- [37] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
- [38] M. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of trust: a decentralized blockchain-based authentication system for IoT," *Computers & Security*, vol. 78, pp. 126–142, 2018.
- [39] M. Shariq, K. Singh, M. Y. Bajuri, A. Pantelous, A. Ahmadian, and M. Salimi, "A secure and reliable RFID authentication protocol using digital schnorr cryptosystem for IoT-enabled healthcare in COVID-19 scenario," *Sustainable Cities and Society*, vol. 75, article 103354, 2021.
- [40] R. Canetti and H. Krawczyk, "Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels," in *Advances in Cryptology|EUROCRYPT 2001*, B. Pfitzmann, Ed., pp. 453–474, Springer, Berlin Heidelberg., 2001.
- [41] R. Canetti and H. Krawczyk, "Universally Composable Notions of Key Exchange and Secure Channels," in *Advances in Cryptology|EURO-CRYPT 2002*, L. R. Knudsen, Ed., pp. 337–351, Springer, Berlin Heidelberg, 2002.
- [42] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K. K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9390–9401, 2020.
- [43] S. Roy, A. K. Das, S. Chatterjee, N. Kumar, S. Chattopadhyay, and J. J. P. C. Rodrigues, "Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 457–468, 2019.
- [44] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE transactions on computers*, vol. 51, no. 5, pp. 541–552, 2002.
- [45] X. Huang, X. Chen, J. Li, Y. Xiang, and L. Xu, "Further observations on smartcard-based password-authenticated key agreement in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1767–1775, 2014.
- [46] B. Yu and H. Li, "Anonymous authentication key agreement scheme with pairingbased cryptography for home-based multi-sensor Internet of Things," *International Journal of Distributed Sensor Networks*, vol. 15, no. 9, 2019.
- [47] M. Shuai, N. Yu, H. Wang, and L. Xiong, "Anonymous authentication scheme for smart home environment with provable security," *Computers and Security*, vol. 86, pp. 132–146, 2019.
- [48] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, and Y. Park, "An efficient, anonymous and robust authentication scheme for smart home environments," *Sensors*, vol. 20, no. 4, p. 1215, 2020.
- [49] J. Zhao, W. Bian, D. Xu et al., "A secure biometrics and PUFs-based authentication scheme with key agreement for multiserver environments," *IEEE Access*, vol. 8, pp. 45292–45303, 2020.

## Research Article

# Provably Secure ECC-Based Three-Factor Authentication Scheme for Mobile Cloud Computing with Offline Registration Centre

Hongwei Luo <sup>1,2</sup>, Feifei Wang <sup>3</sup>, and Guoai Xu <sup>1,2</sup>

<sup>1</sup>Beijing University of Posts and Telecommunications, Beijing 100876, China

<sup>2</sup>National Engineering Laboratory of Mobile Network Security, Beijing 100876, China

<sup>3</sup>Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Correspondence should be addressed to Guoai Xu; [xga@bupt.edu.cn](mailto:xga@bupt.edu.cn)

Received 6 August 2020; Revised 22 December 2020; Accepted 9 May 2021; Published 29 May 2021

Academic Editor: Weizhi Meng

Copyright © 2021 Hongwei Luo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile cloud computing (MCC) aims at solving the resource constrain problem of smart mobile devices. It has deeply affected the way modern humans live and work. In MCC, the authentication scheme is indispensable to prevent illegal attacks and privacy breaches. In this paper, we reveal that a recently proposed two-factor authentication scheme for MCC has limitations like stolen-verifier attack and denial of service attack. In addition, its single-server architecture is not applicable to MCC. To enhance the security, we present a provably secure three-factor authentication scheme using the elliptic curve cryptosystem (ECC). It has the merit that the user only needs to register once to access multiple servers with a pair of public and private key, and the registration center is offline in the authentication phase. Security analysis demonstrates that our scheme is immune to known attacks and provides user friendliness. Finally, performance comparisons indicate that our scheme has better security attributes and low computing and communication overheads, and it is more applicable to MCC.

## 1. Introduction

With the popularity of smart mobile devices, mobile Internet is becoming more and more important in our daily life and deeply affects the way modern humans live and work [1]. Mobile Internet provides high-quality telecommunication services such as voice, fax, data, image, and multimedia. We can obtain a variety of services anytime and anywhere through mobile Internet. Various mobile Internet applications include mobile payment, mobile e-commerce, and mobile entertainment are emerged. Some of these applications such as WeChat and Alipay bring tremendous convenience to people. With the continuous development of mobile Internet, the deficiency that smart mobile devices have limited storage capacity and processing power is gradually revealed. To resolve this issue, cloud computing [2] is introduced into mobile Internet; therefore, a new technology namely mobile cloud computing (MCC) [3] is produced. It

aims at solving the resource constrain problem of smart mobile devices, and it can effectively increase the computing power and storage capacity of smart mobile devices.

In an MCC setting, as a trusted third party, the registration center is responsible for issuing the secret key to users and cloud servers in the registration phase. In the authentication phase, the users access the resources and services deployed in distributed cloud servers via mobile and wireless networks, as shown in Figure 1. Due to the openness of the communication networks, the attacker can implement various attacks such as modification, forgery, and replay. It is indispensable to develop an authentication scheme for MCC to achieve identity authentication and secure data transmission, as well as the protection of user privacy.

*1.1. Related Works.* Since Lamport [4] presented the first password authentication scheme, a large number of schemes [5–18] that are applicable to different scenarios, adopt differ-

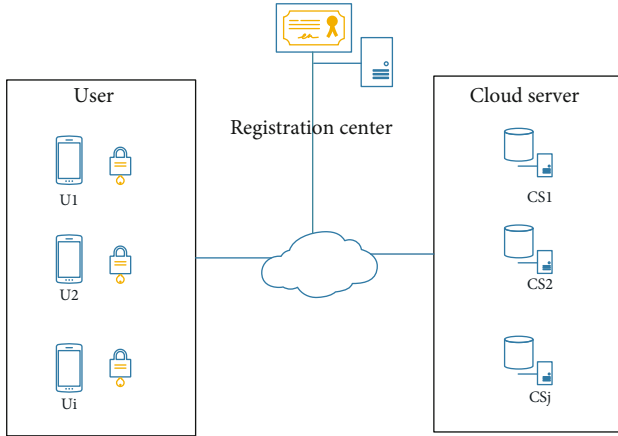


FIGURE 1: The architecture of MCC.

ent cryptosystems, and employ different kinds of authentication factors were presented. In 2001, Li et al. [17] presented the first multiserver authentication scheme, in which the user can register once and then access multiple servers with a pair of identity and password. Some authentication schemes for MCC [19–22] have been presented in recent years. In 2015, Tsai and Lo [3] introduced an authentication scheme for MCC with offline registration center using bilinear pairing. In 2017, Feng et al. [23] introduced a three-factor mobile multiserver authentication scheme using the elliptic curve cryptosystem (ECC). Amin et al. [24] introduced a lightweight two-factor authentication scheme for MCC. However, their scheme is found to have weaknesses such as offline guessing attack [25]. In 2018, He et al. [26] pointed out that Tsai et al.’s scheme suffers from server impersonation attack. They furthermore proposed an improved scheme by using identity-based signature. Their scheme can provide better security features. In 2019, Irshad et al. [27] presented an enhanced authentication scheme for MCC using bilinear pairing. In 2019, Mo et al. [28] put forward a provably secure two-factor authentication scheme using ECC. In 2020, Li et al. [29] put forward a lattice-based password authenticated key exchange protocol, and their scheme achieves quantum resistance.

**1.2. Motivation and Contributions.** To improve the security and optimize the efficiency, we design a provably secure authentication scheme using ECC in this paper. Without public key cryptographic techniques, it is difficult to achieve user anonymity and forward secrecy [12]. By using ECC, the proposed scheme provides mutual authentication and user anonymity and establishes secure session key. Compared with the existing schemes with offline registration center using bilinear pairing [3, 26–28], our ECC-based scheme is more efficient. Our major contributions are as follows.

- (1) We prove that Mo et al.’s scheme [28] has limitations like stolen-verifier attack, denial of service attack, known session-specific temporary information attack, and its single-server architecture is not applicable to MCC

- (2) We put forward a novel authentication scheme for MCC using ECC. It inherits the advantages of existing schemes such as He et al.’s scheme. It enables the user to register once and use a pair of public and private key to access multiple servers. In the authentication phase, the registration center is offline. The user interacts with the cloud server directly. It is conducive to reduce computing and communication overheads
- (3) The security analysis demonstrates that the proposed scheme can resist usual attacks and preserve user friendliness. The performance comparisons show that the proposed scheme can remedy the security defects of the existing schemes and incur low computing and communication overheads. The proposed scheme is more suitable for MCC

**1.3. Roadmap of Paper.** This paper is organized as below. Section 2 gives some preliminaries. Mo et al.’s scheme is cryptanalyzed in Section 3. Section 4 gives the proposed three-factor authentication scheme for MCC. Section 5 is the security analysis. Section 6 is the performance comparisons. Finally, we conclude the paper in Section 7. We summarize some notations in Table 1.

## 2. Preliminaries

**2.1. Elliptic Curve Diffie-Hellman Problem.** Elliptic curve Diffie-Hellman problem (ECDHP):  $E_q$  is an elliptic curve group over the prime field  $F_p$ ,  $P$  is a generator of  $E_q$ . For given  $\alpha P, \beta P$ , where  $\alpha, \beta \in Z_q^*$ , solving  $\alpha\beta P$  is intractable [30].

**2.2. Adversary Model.** In the light of [31], we suppose that the ability of attacker is as below.

- (i) We suppose that the attacker can block, modify, and eavesdrop the message delivered via the public channel
- (ii) We suppose that the attacker is able to enumerate all pairs of identity and password subordinate to the dictionary space
- (iii) We suppose that the attacker can compromise one type of authentication factor of user, i.e., smart card, password, or biometric
- (iv) When evaluating three-factor secrecy, we suppose that the attacker can compromise any two types of authentication factors

## 3. Analysis of Mo Et al.’s Scheme

**3.1. Review of Mo Et al.’s Scheme.** We briefly describe Mo et al.’s two-factor single-server authentication scheme for MCC [28] in this section. To initialize the system, the cloud server CS selects the master key  $s$  and calculates the public key  $PUB = sP$ .

**3.1.1. User Registration Phase.** This phase is executed as follows.



TABLE 1: Notations.

Symbols	Description
RC	The registration center
$U_i$	The user
$CS_j$	The cloud server
$ID_i, PW_i, b_i$	$U_i$ 's identity, password, and biometric
$SID_j$	$CS_j$ 's identity
$P$	A generator of elliptic curve group $E_q$
$d_i, PUB_i$	$U_i$ 's private key and public key
$k_j, PUB_j$	$CS_j$ 's private key and public key
SK	Session key
	The string concatenation operation
$\oplus$	The bitwise XOR operation
$H_1()$	Hash function
$H_2()$	Biohashing function, it maps the biometric of user to a random string

(Step1)  $U_i \rightarrow CS : \{ID_i, R_i\}$ . The user  $U_i$  selects his identity  $ID_i$ , password  $PW_i$ , and a nonce  $r_i$  and computes  $R_i = H_1(r_i || PW_i)$ .

(Step2)  $CS \rightarrow U_i$ : a smart card. CS picks a nonce  $N_i$  and computes  $F_i = H_1(H_1(ID_i || N_i || T_i || SC_i) \bmod \nu)$ ,  $A_i = F_i \oplus R_i$ , where  $T_i$  is the current timestamp,  $\nu$  is an integer from  $[2^4, 2^8]$ , and  $SC_i$  is the smart card identification number. CS stores  $(ID_i, N_i, T_i, SC_i)$  in the database and stores  $\{A_i, ID_S, PUB, \nu\}$  in a smart card, where  $ID_S$  is the identity of CS

(Step3)  $U_i$  computes  $B_i = r_i \oplus H_1(ID_i || R_i) \bmod \nu$  and stores  $B_i$  in the smart card

The user  $U_i$  selects his identity  $ID_i$ , password  $PW_i$ , and a nonce  $r_i$  and computes  $R_i = H_1(r_i || PW_i)$ .

**3.1.2. Authentication Phase.** This phase is comprised of the following steps.

(Step1)  $U_i \rightarrow CS : \{PID_i, C_i, L_i\}$ .  $U_i$  enters  $ID_i^*, PW_i^*$ . Then, the smart card computes  $R_i^* = H_1(r_i || PW_i^*)$ ,  $B_i^* = r_i \oplus H_1(ID_i^* || R_i^*) \bmod \nu$  and checks if  $B_i^* = B_i$ . If it holds, the smart card chooses a nonce  $r_1$  and computes  $C_i = r_1 P$ ,  $D_i = r_1 PUB$ ,  $E_i = C_i + D_i$ ,  $F_i = A_i \oplus R_i$ , the dynamic identity  $PID_i = (ID_i^* || F_i) \oplus H_1(C_i || E_i)$ , and  $L_i = H_1(ID_i^* || D_i || PID_i)$ .

(Step2)  $CS \rightarrow U_i : \{M_1, M_3\}$ . CS computes  $D_i = sC_i$ ,  $E_i = C_i + D_i$ ,  $(ID_i || F_i) = PID_i \oplus H_1(C_i || E_i)$ , and  $L_i^* = H_1(ID_i || D_i || PID_i)$  and checks if  $L_i^* = L_i$ . If it does not hold, the protocol aborts. Otherwise, CS retrieves  $(ID_i, N_i, T_i, SC_i)$  from the database based on  $ID_i$  and computes  $F_i^* = H_1(H_1(ID_i || N_i || T_i || SC_i) \bmod \nu)$  and checks if  $F_i^* = F_i$ . If they

are equal, CS chooses a nonce  $r_2$  and computes  $M_1 = r_2 P$ ,  $M_2 = r_2 C_i$ , the session key  $SK = H_1(ID_i || ID_S || D_i || M_1 || M_2)$ , and  $M_3 = H_1(ID_i || ID_S || C_i || D_i || M_1 || M_2)$ .

(Step3)  $U_i \rightarrow CS : \{M_4\}$ .  $U_i$  computes  $M_2 = r_1 M_1$ ,  $M_3^* = H_1(ID_i || ID_S || C_i || D_i || M_1 || M_2)$ , and checks if  $M_3^* = M_3$ . If they are equal,  $U_i$  computes  $SK = H_1(ID_i || ID_S || D_i || M_1 || M_2)$  and  $M_4 = H_1(ID_i || ID_S || D_i || M_2 || SK)$ .

(Step4) CS computes  $M_4^* = H_1(ID_i || ID_S || D_i || M_2 || SK)$  and checks if  $M_4^* = M_4$ . If they are not equal, the protocol aborts

$U_i$  enters  $ID_i^*, PW_i^*$ . Then, the smart card computes  $R_i^* = H_1(r_i || PW_i^*)$ ,  $B_i^* = r_i \oplus H_1(ID_i^* || R_i^*) \bmod \nu$  and checks if  $B_i^* = B_i$ . If it holds, the smart card chooses a nonce  $r_1$  and computes  $C_i = r_1 P$ ,  $D_i = r_1 PUB$ ,  $E_i = C_i + D_i$ ,  $F_i = A_i \oplus R_i$ , the dynamic identity  $PID_i = (ID_i^* || F_i) \oplus H_1(C_i || E_i)$ , and  $L_i = H_1(ID_i^* || D_i || PID_i)$ .

**3.1.3. Smartcard Revocation Phase.** The smart card can be revoked through the following steps.

(Step1)  $U_i$  performs step 1 of the authentication phase.  $U_i$  sends a revocation request  $\{PID_i, C_i, L_i, revoke\_request\}$  to CS

(Step2) CS checks if  $L_i^* = L_i$  and  $F_i^* = F_i$ . If they are equal, CS deletes  $(ID_i, N_i, T_i, SC_i)$  from the database

Performs step 1 of the authentication phase.  $U_i$  sends a revocation request  $\{PID_i, C_i, L_i, revoke\_request\}$  to CS

After that, the smart card cannot be used to login CS. The user reregisters with CS to get a new smart card.

**3.2. Weaknesses of Mo et al.'s Scheme.** In this section, we prove that Mo et al.'s scheme is not immune to various attacks.

**3.2.1. Stolen-Verifier Attack.** In Mo et al.'s scheme, CS stores a tuple  $(ID_i, N_i, T_i, SC_i)$  for each user  $U_i$ . If the attacker compromises CS and retrieves  $(ID_i, N_i, T_i, SC_i)$  from the database, the attacker can masquerade as the legitimate user through the following steps.

(Step1) The attacker computes  $F_i = H_1(H_1(ID_i || N_i || T_i || SC_i) \bmod \nu)$

(Step2) The attacker chooses a nonce  $r_1$  and computes  $C_i = r_1 P$ ,  $D_i = r_1 PUB$ ,  $E_i = C_i + D_i$ ,  $PID_i = (ID_i || F_i) \oplus H_1(C_i || E_i)$ ,  $L_i = H_1(ID_i || D_i || PID_i)$ .  $U_i$  sends  $\{PID_i, C_i, L_i\}$  to CS

As  $L_i^* = L_i$  and  $F_i^* = F_i$ , CS regards the attacker as the legitimate user  $U_i$ . The essential reason for this attack is that the secret authentication value  $F_i$  is merely based on the information stored in verification table, rather than the secret key of CS.

3.2.2. *Denial of Service Attack.* This attack is performed as follows.

(Step1) The adversary intercepts  $\{PID_i, C_i, L_i\}$  from the public channel

(Step2) The attacker sends  $\{PID_i, C_i, L_i, revoke\_request\}$  to CS

After receiving  $\{PID_i, C_i, L_i, revoke\_request\}$ , as it is valid, CS deletes  $(ID_i, N_i, T_i, SC_i)$  from the database. After that, the legitimate user  $U_i$  is unable to access CS unless reregistration. The essential reason for this attack is that CS does not check the freshness of  $\{PID_i, C_i, L_i, revoke\_request\}$ . The attacker can forge a revocation request using the intercepted  $\{PID_i, C_i, L_i\}$ .

3.2.3. *Known Session-Specific Temporary Information Attack.* Once the attacker compromises the nonce  $r_1$ , he can reveal the session key through the following steps.

(Step1) The attacker intercepts  $\{PID_i, C_i, L_i\}$  and  $\{M_1, M_3\}$  from the public channel

(Step2) The attacker obtains the user identity by shoulder peeping or computing  $D_i = r_1 \text{PUB}$ ,  $E_i = C_i + D_i$ ,  $(ID_i \| F_i) = PID_i \oplus H_1(C_i \| E_i)$

(Step3) The attacker can obtain  $ID_s$  by compromising user's smart card or colluding with a user

(Step4) The attacker computes  $M_2 = r_1 M_1$ ,  $SK = H_1(ID_i \| ID_s \| D_i \| M_1 \| M_2)$ .

3.2.4. *Not Applicable to Mobile Cloud Computing.* Mo et al.'s scheme adopts single server architecture. Only a single server is used to handle the access requests of users. However, in the MCC environment, a large number of users access the cloud server to obtain a variety of services using mobile devices. It is impracticable for a single server to deal with all the access requests in time. MCC aims at integrating the resources and computing power of multiple distributed servers. As depicted in Figure 1, the MCC architecture usually involves multiple distributed servers. In Mo et al.'s scheme, its single-server architecture is not applicable to MCC.

## 4. The Proposed Scheme

In this section, we put forward an ECC-based three-factor authentication scheme for MCC. It includes three kinds of participants, i.e., the registration center RC, the cloud server  $CS_j$ , and the user  $U_i$ . As a trusted third party, RC is responsible for issuing the secret key to users and cloud servers in the registration phase. In the authentication phase, RC is offline.  $U_i$  and  $CS_j$  implement mutual authentication and negotiate a session key without the registration center involved.

4.1. *Predeployment Phase.* RC selects an elliptic curve group  $E_q$  over the prime field  $F_p$ .  $P$  is a generator of  $E_q$ . RC selects the master key  $s$ . RC chooses a secure hash function  $H_1()$

and a bihashing function  $H_2()$ . RC publishes the parameters  $\{E_q, P\}$ .

4.2. *User Registration Phase.* This phase is depicted as Figure 2.

(Step1) The user  $U_i$  chooses his identity  $ID_i$  and password  $PW_i$ , imprints his biometric  $b_i$ , and computes  $RPW_i = H_1(ID_i \| PW_i \| H_2(b_i) \| y_i)$ , where  $y_i$  is a nonce.  $U_i$  delivers the message  $\{ID_i, RPW_i\}$  to RC via the reliable channel

(Step2) After getting  $\{ID_i, RPW_i\}$ , RC computes  $U_i$ 's private key  $d_i = H_1(ID_i \| s \| RPW_i)$  and public key  $\text{PUB}_i = d_i P$ ,  $W_i = d_i \oplus RPW_i$ ,  $Z_i = H_1(RPW_i) \bmod v$ . RC chooses an integer  $v \in [2^4, 2^8]$ . RC stores the parameters  $\{W_i, Z_i, v\}$  in a smart card and publishes  $U_i$ 's public key  $\{ID_i, \text{PUB}_i\}$ . RC issues the smart card to  $U_i$  in a credible manner

(Step3)  $U_i$  saves  $y_i$  in the smart card

4.3. *Cloud Server Registration Phase.* This phase is depicted as Figure 3.

(Step1) The cloud server  $CS_j$  delivers his identity  $\{SID_j\}$  to RC via the reliable channel

(Step2) Upon getting  $\{SID_j\}$ , RC computes  $CS_j$ 's private key  $k_j = H_1(SID_j \| s)$  and public key  $\text{PUB}_j = k_j P$ . RC publishes the parameters  $\{SID_j, \text{PUB}_j\}$ . RC issues  $\{k_j\}$  to  $CS_j$  in a credible manner

4.4. *Authentication Phase.* This phase is depicted as Figure 4.

(Step1)  $U_i$  enters  $ID_i^*$  and  $PW_i^*$  and imprints  $b_i^*$ . The smart card computes  $RPW_i^* = H_1(ID_i^* \| PW_i^* \| H_2(b_i^*) \| y_i)$ ,  $Z_i^* = H_1(RPW_i^*) \bmod v$ , and checks if  $Z_i^* = Z_i$ . If they are equal, the smart card chooses two random numbers  $r_1$  and  $r_2$  and computes  $d_i = W_i \oplus RPW_i^*$ ,  $A_i = r_1 P$ ,  $B_i = r_1 \text{PUB}_j$ ,  $N_i = r_2 P$ ,  $C_i = H_1(A_i \| ID_i \| N_i)$ ,  $D_i = r_1 + d_i$ ,  $E_i = B_i \oplus (ID_i \| D_i \| N_i)$ .  $U_i$  sends the message  $\{A_i, N_i, E_i\}$  to  $CS_j$  via the public channel

(Step2) Upon receiving  $\{A_i, N_i, E_i\}$ ,  $CS_j$  computes  $B_i = k_j A_i$ ,  $(ID_i \| D_i \| N_i) = E_i \oplus B_i$ ,  $C_i = H_1(A_i \| ID_i \| N_i)$  and checks if  $D_i P = A_i + C_i \cdot \text{PUB}_j$ . If it holds,  $CS_j$  chooses a random number  $r_3$  and computes  $F_i = r_3 P$ , the session key  $SK = H_1(r_3 N_i \| D_i)$ ,  $L_i = H_1(SK \| F_i)$ .  $CS_j$  sends  $\{F_i, L_i\}$  to  $U_i$

(Step3) After receiving  $\{F_i, L_i\}$ , the smart card computes  $SK = H_1(r_2 F_i \| D_i)$ ,  $L_i^* = H_1(SK \| F_i)$  and verifies if  $L_i^* = L_i$ . If so, the smart card computes  $M_i = H_1(SK \| B_i)$ .  $U_i$  sends  $\{M_i\}$  to  $CS_j$

(Step4) Upon getting  $\{M_i\}$ ,  $CS_j$  computes  $M_i^* = H_1(SK \| B_i)$  and checks if  $M_i^* = M_i$ . If they are equal,

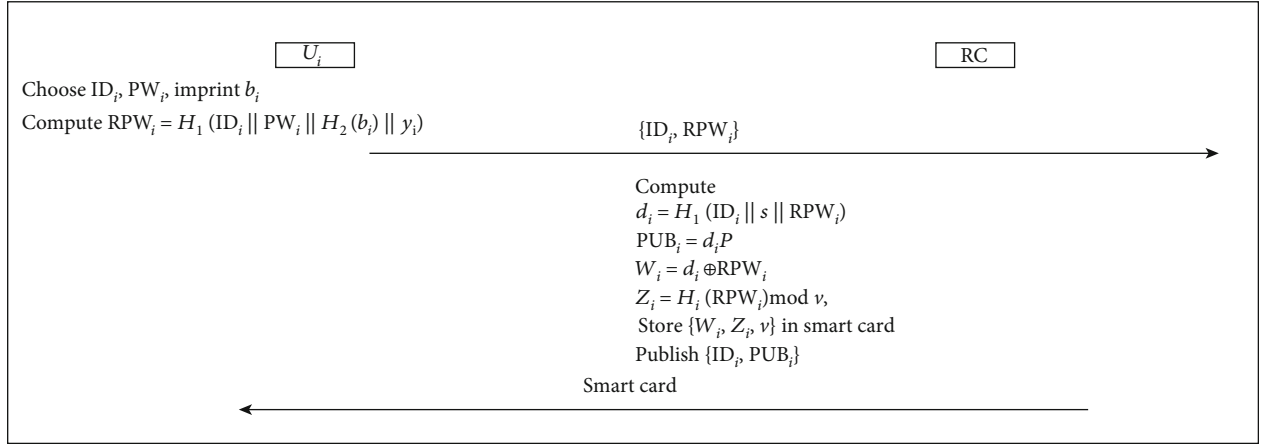


FIGURE 2: User registration phase of the proposed scheme.

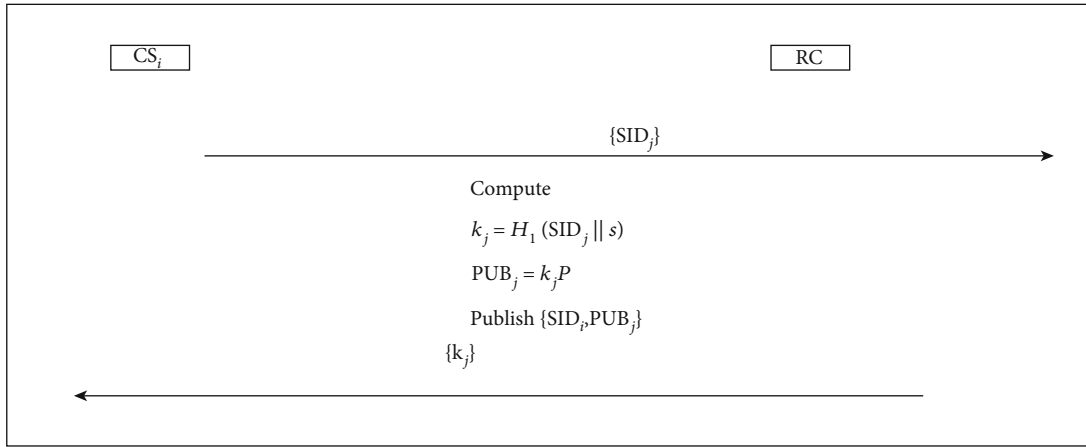


FIGURE 3: Cloud server registration phase of the proposed scheme.

CS<sub>j</sub> and U<sub>i</sub> achieve mutual authentication and establish a session key SK

**4.5. Smart Card Revocation Phase.** If user's smart card is lost or stolen, the user suspects that the data of smart card is leaked. The user reregisters with RC. RC publishes user's new public key information {ID<sub>i</sub>, PUB<sub>i</sub><sup>new</sup>} and issues a new smart card to U<sub>i</sub>. Afterwards, the user's old smart card is unable to be used to login any cloud server.

**4.6. Password and Biometric Update Phase.** This phase is executed as follows.

- (Step1) U<sub>i</sub> inputs ID<sub>i</sub><sup>\*</sup> and PW<sub>i</sub><sup>\*</sup> and imprints b<sub>i</sub><sup>\*</sup>. The smart card computes RPW<sub>i</sub><sup>\*</sup> = H<sub>1</sub>(ID<sub>i</sub><sup>\*</sup> || PW<sub>i</sub><sup>\*</sup> || H<sub>2</sub>(b<sub>i</sub><sup>\*</sup>) || y<sub>i</sub>), Z<sub>i</sub><sup>\*</sup> = H<sub>1</sub>(RPW<sub>i</sub><sup>\*</sup>) mod v and checks if Z<sub>i</sub><sup>\*</sup> = Z<sub>i</sub>. If they are equal, ask the user to input his new password and imprint his new biometric
- (Step2) The smart card chooses a new nonce y<sub>i</sub><sup>new</sup> and computes RPW<sub>i</sub><sup>new</sup> = H<sub>1</sub>(ID<sub>i</sub><sup>\*</sup> || PW<sub>i</sub><sup>new</sup> || H<sub>2</sub>(b<sub>i</sub><sup>new</sup>) || y<sub>i</sub><sup>new</sup>), Z<sub>i</sub><sup>new</sup> = H<sub>1</sub>(RPW<sub>i</sub><sup>new</sup>) mod v, and W<sub>i</sub><sup>new</sup> =

W<sub>i</sub> ⊕ RPW<sub>i</sub><sup>\*</sup> ⊕ RPW<sub>i</sub><sup>new</sup>. The smart card saves W<sub>i</sub><sup>new</sup>, Z<sub>i</sub><sup>new</sup> and deletes W<sub>i</sub>, Z<sub>i</sub>

U<sub>i</sub> inputs ID<sub>i</sub><sup>\*</sup> and PW<sub>i</sub><sup>\*</sup> and imprints b<sub>i</sub><sup>\*</sup>. The smart card computes RPW<sub>i</sub><sup>\*</sup> = H<sub>1</sub>(ID<sub>i</sub><sup>\*</sup> || PW<sub>i</sub><sup>\*</sup> || H<sub>2</sub>(b<sub>i</sub><sup>\*</sup>) || y<sub>i</sub>), Z<sub>i</sub><sup>\*</sup> = H<sub>1</sub>(RPW<sub>i</sub><sup>\*</sup>) mod v and checks if Z<sub>i</sub><sup>\*</sup> = Z<sub>i</sub>. If they are equal, ask the user to input his new password and imprint his new biometric

## 5. Security Analysis

In this section, we prove the security of the proposed scheme by using the following security analysis methods.

**5.1. BAN Logic Proof.** In this section, we show that the proposed scheme preserves mutual authentication and session key agreement by using BAN logic proof. We present the notations and rules of BAN logic [32] in Table 2.

The proposed scheme should be able to achieve the following goals.

- G1: U<sub>i</sub> |≡ CS<sub>j</sub> |≡ (U<sub>i</sub>  $\stackrel{\text{SK}}{\leftrightarrow}$  CS<sub>j</sub>)  
 G2: U<sub>i</sub> |≡ (U<sub>i</sub>  $\stackrel{\text{SK}}{\leftrightarrow}$  CS<sub>j</sub>)  
 G3: CS<sub>j</sub> |≡ U<sub>i</sub> |≡ (U<sub>i</sub>  $\stackrel{\text{SK}}{\leftrightarrow}$  CS<sub>j</sub>)

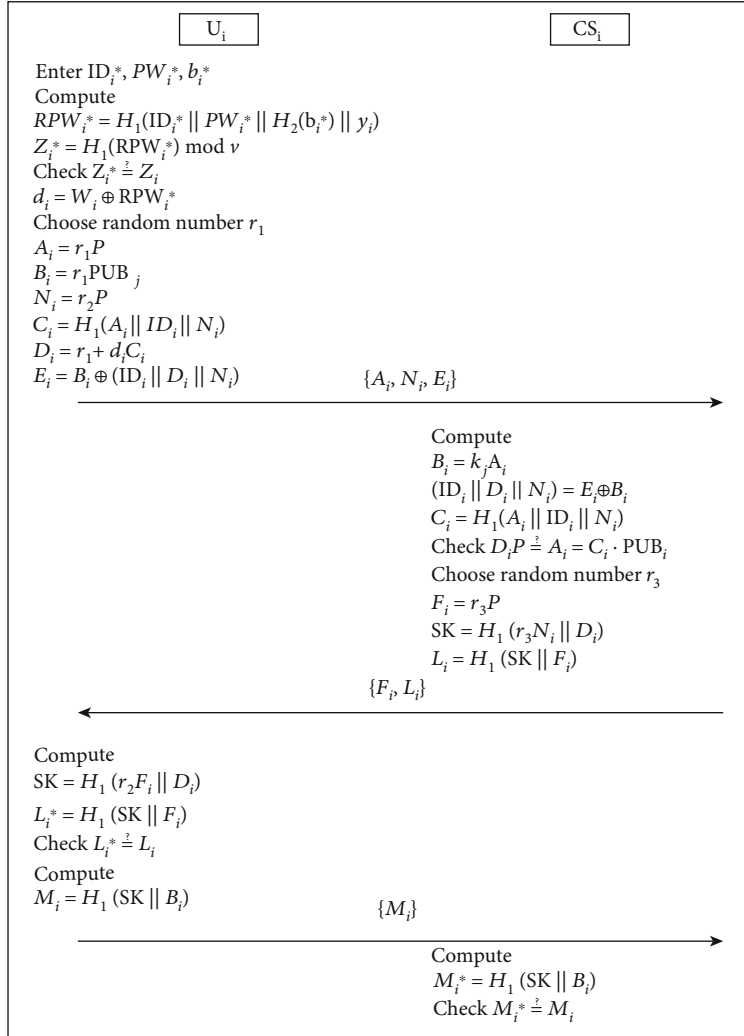


FIGURE 4: Authentication phase of the proposed scheme.

$$G4: CS_j | \equiv (U_i \stackrel{SK}{\leftrightarrow} CS_j)$$

The proposed scheme is idealized as below.

$$M1: U_i \rightarrow CS_j \{D_i = \{ID_i, r_1 P, r_2 P\}_{d_i}, r_2 P\}_{B_i}$$

$$M2: CS_j \rightarrow U_i \langle U_i \stackrel{SK}{\leftrightarrow} CS_j, r_3 P \rangle_{D_i}$$

$$M3: U_i \rightarrow CS_j \langle U_i \stackrel{SK}{\leftrightarrow} CS_j, r_1 P \rangle_{B_i}$$

The initial assumptions of the proposed scheme are as below.

$$A1: CS_j | \equiv U_i \stackrel{B_i}{\leftrightarrow} CS_j$$

$$A2: CS_j | \equiv \xrightarrow{PUB_i} U_i$$

$$A3: CS_j | \equiv \#(r_1 P)$$

$$A4: CS_j | \equiv U_i \Rightarrow r_2 P$$

$$A5: U_i | \equiv U_i \stackrel{D_i}{\leftrightarrow} CS_j$$

$$A6: U_i | \equiv \#(r_3 P)$$

$$A7: U_i | \equiv CS_j \Rightarrow U_i \stackrel{SK}{\leftrightarrow} CS_j$$

$$A8: CS_j | \equiv U_i \Rightarrow U_i \stackrel{SK}{\leftrightarrow} CS_j$$

The proof is as follows.

From M1, we have

$$(1) CS_j \triangleleft \{D_i, r_2 P\}_{B_i}$$

Apply Rule 1 to (1) and A1, we have

$$(2) CS_j | \equiv U_i | \sim (D_i, r_2 P)$$

From (2), we have

$$(3) CS_j | \equiv U_i | \sim D_i (\{ID_i, r_1 P, r_2 P\}_{d_i})$$

Apply Rule 1 to (3) and A2, we have

$$(4) CS_j | \equiv U_i | \sim (ID_i, r_1 P, r_2 P)$$

Apply Rule 2 to (4) and A3, we have

$$(5) CS_j | \equiv U_i | \equiv (ID_i, r_2 P)$$

Apply Rule 3 to (5) and A4, we have

$$(6) CS_j | \equiv r_2 P$$

From M2, we have

$$(7) U_i \triangleleft \langle U_i \stackrel{SK}{\leftrightarrow} CS_j, r_3 P \rangle_{D_i}$$

Apply Rule 1 to (7) and A5, we have

$$(8) U_i | \equiv CS_j | \sim (U_i \stackrel{SK}{\leftrightarrow} CS_j, r_3 P)$$

Apply Rule 2 to (8) and A6, we have

$$(9) U_i | \equiv CS_j | \equiv U_i \stackrel{SK}{\leftrightarrow} CS_j (G1)$$

Apply Rule 3 to (9) and A7, we have

TABLE 2: The notations and rules of BAN logic.

Symbols	Description
$P, Q$	A principal
$X$	A statement
$\#(X)$	$X$ is fresh
$P \triangleleft X$	$P$ gets $X$
$P   \sim X$	$X$ is sent by $P$
$P   \equiv X$	$P$ believes $X$
$P \stackrel{K}{\leftrightarrow} Q$	$P$ and $Q$ have a common secret $K$
$\{X\}_K$	$X$ is encrypted under $K$
$\stackrel{K}{\rightarrow} P$	$K$ is the public key of $P$
$P \Rightarrow X$	$P$ has jurisdiction over $X$
$\langle X \rangle_K$	$X$ is merged with $K$
Message meaning rule (rule 1)	$\frac{P   \equiv P \stackrel{K}{\leftrightarrow} Q, P \triangleleft \langle X \rangle_K}{P   \equiv Q   \sim X}$ or $\frac{P   \equiv \rightarrow Q, P \triangleleft \{X\}_{K^{-1}}}{P   \equiv Q   \sim X}$
Nonce-verification rule (rule 2)	$\frac{P   \equiv \#(X), P   \equiv Q   \sim X}{P   \equiv Q   \equiv X}$
Jurisdiction rule (rule 3)	$\frac{P   \equiv Q \Rightarrow X, P   \equiv Q   \equiv X}{P   \equiv X}$

$$(10) U_i | \equiv U_i \stackrel{SK}{\leftrightarrow} CS_j(G2)$$

From M3, we have

$$(11) CS_j \triangleleft U_i \stackrel{SK}{\leftrightarrow} CS_j, r_1 P >_{B_i}$$

Apply Rule1 to (11) and A1, we have

$$(12) CS_j | \equiv U_i | \sim (U_i \stackrel{SK}{\leftrightarrow} CS_j, r_1 P)$$

Apply Rule 2 to (14) and A3, we have

$$(13) CS_j | \equiv U_i | \equiv (U_i \stackrel{SK}{\leftrightarrow} CS_j)(G3)$$

Apply Rule 3 to (15) and A8, we have

$$(14) CS_j | \equiv (U_i \stackrel{SK}{\leftrightarrow} CS_j)(G4)$$

**5.2. Formal Security Analysis.** In this section, we show that the proposed scheme is provably secure under the security model introduced in [33].

### 5.2.1. Security Model

(1) *Participants.* The proposed scheme involves three kinds of participants, i.e., the registration center RC, the cloud server  $CS_j$ , and the user  $U_i$ .  $RC^a$ ,  $CS_j^a$ , and  $U_i^a$  are the  $a$ -th instances of RC,  $CS_j$ , and  $U_i$ , respectively.

(2) *Queries.* The adversary capability is simulated through the following queries.

Execute  $(CS_j^a/U_i^a)$ . It simulates the passive attack. It returns back the transcript of messages to the adversary.

Send  $(CS_j^a/U_i^a, m)$ . It simulates the active attack. The adversary masquerades as the instance  $CS_j^a/U_i^a$  by sending a message  $m$ . The oracle processes  $m$  and returns a response to the adversary.

Reveal  $(CS_j^a/U_i^a)$ . It returns back  $CS_j^a/U_i^a$ 's session key to the adversary.

Corrupt  $(U_i^a, z)$ . It returns back one or two kinds of user authentication factors to the adversary.

If  $z = 1$ , it returns back the password.

If  $z = 2$ , it returns back the data of smart card.

If  $z = 3$ , it returns back the biometric.

Corrupt  $(RC^a/CS_j^a)$ . It simulates forward secrecy. The oracle returns back the master key of  $RC^a$  or the private key of  $CS_j^a$  to the adversary.

Test  $(CS_j^a/U_i^a)$ . It simulates the semantic security of the session key, If the instance  $CS_j^a/U_i^a$  is accepted by its partner and establishes a session key SK, and the adversary never makes Corrupt  $(RC^a/CS_j^a)$  or Reveal  $(CS_j^a/U_i^a)$  query, we say the instance  $CS_j^a/U_i^a$  is fresh. If  $CS_j^a/U_i^a$  is fresh, the oracle tosses a coin  $b$ . If  $b = 1$ , it answers SK. Otherwise, it chooses an equal-length string and sends it to the adversary. The adversary is allowed to make this query no more than once.

(3) *Semantic Security.* After receiving the answer from Test  $(CS_j^a/U_i^a)$  query, the adversary tries to reveal the value of  $b$ . We define the advantage that adversary breaks the semantic security of the proposed scheme as

$$Adv_P^{ake}(\mathcal{A}) = 2 \Pr(b' = b) - 1. \quad (1)$$

If  $Adv_P^{ake}(\mathcal{A})$  is negligible, the proposed scheme achieves semantic security.

### 5.2.2. Security Analysis

**Theorem 1.** As demonstrated in [34], the password distribution follows Zipf's law.  $|D_{PW}|$  denotes the password dictionary space.  $C'$  and  $s'$  are parameters of the Zipf distribution.  $Adv_P^{ECDHP}$  denotes the advantage that the adversary  $\mathcal{A}$  solves ECDHP. The adversary  $\mathcal{A}$  can make at most  $q_e$  Execute queries,  $q_s$  Send queries,  $q_h$  Hash queries, and  $q_b$  Biohashing queries in polynomial time  $t$ . We have

$$Adv_P^{ake}(\mathcal{A}) \leq 2C' * q_s^{s'} + \frac{(q_s + q_e)^2}{p} + \frac{6q_s + q_h^2}{2^{l_1}} + \frac{2q_s + q_b^2}{2^{l_2}} + 2q_h Adv_P^{ECDHP}, \quad (2)$$

where  $l_1$  is the length of the hash value, and  $l_2$  is the length of the biohashing value, in terms of the Tianya password dictionary [35] of size  $|D_{PW}| \approx 13$  million,  $C' = 0.062239$ ,  $s' = 0.155478$ .

Proof. The security of the proposed scheme is demonstrated through a series of games  $\Phi_i$  ( $0 \leq i \leq 6$ ), and  $\Pr[\chi_i]$  denotes the advantage that  $\mathcal{A}$  guesses  $b$  in  $\Phi_i$ .

$\Phi_0$ : this game represents the real attack. Hence,

$$\text{Adv}_P^{\text{ake}}(\mathcal{A}) = 2(\Pr[\chi_0]) - 1. \quad (3)$$

$\Phi_1$ : the hash oracle and biohashing oracle are simulated by setting up two lists  $\Lambda_H$  and  $\Lambda_{BH}$ . For a Hash query  $H_1(\tau)$ , the oracle uses  $\tau$  to search  $\Lambda_H$ . If an item  $(\tau, \gamma)$  is found, it sends back  $\gamma$  to the adversary. Otherwise, it returns a random number  $\gamma$  to the adversary and adds a new item  $(\tau, \gamma)$  to  $\Lambda_H$ . The biohashing oracle is simulated in the same way. There is no difference between  $\Phi_1$  and  $\Phi_0$ . Hence,

$$\Pr[\chi_1] - \Pr[\chi_0] = 0. \quad (4)$$

$\Phi_2$ : This game is terminated when some collisions occur.

- (1) A collision appears in random numbers. The probability is no more than  $(q_s + q_e)^2/2p$
- (2) A collision appears in hash values or biohashing values. The probability is no more than  $q_h^2/2^{l_1+1} + q_b^2/2^{l_2+1}$

Hence,

$$|\Pr[\chi_2] - \Pr[\chi_1]| \leq \frac{q_h^2}{2^{l_1+1}} + \frac{q_b^2}{2^{l_2+1}} + \frac{(q_s + q_e)^2}{2p}. \quad (5)$$

$\Phi_3$ : we abort the game when  $\mathcal{A}$  has guessed  $(D_i, L_i, M_i)$ . Its advantage is no more than  $q_s/2^{l_1}$ . Hence,

$$|\Pr[\chi_3] - \Pr[\chi_2]| \leq q_s/2^{l_1}. \quad (6)$$

$\Phi_4$ : we abort the game when  $\mathcal{A}$  has guessed user's secret key  $d_i$ . Its advantage is no more than  $q_s/2^{l_1}$ . Hence,

$$|\Pr[\chi_4] - \Pr[\chi_3]| \leq q_s/2^{l_1}. \quad (7)$$

$\Phi_5$ : we abort the game when  $\mathcal{A}$  has computed  $d_i$  having the aid of Corrupt  $(U_i^a, z)$  query.

- (1) If  $\mathcal{A}$  has obtained user's password and biometric, he is able to reveal the key parameter  $W_i$  with probability  $q_s/2^{l_1}$
- (2) If  $\mathcal{A}$  has obtained user's password and the data of smart card, he is able to reveal the biometric with probability  $q_s/2^{l_2}$
- (3) If  $\mathcal{A}$  has obtained user's biometric and the data of smart card, he is able to reveal the password with probability  $C' * q_s^{s'}$

Hence,

$$|\Pr[\chi_5] - \Pr[\chi_4]| \leq q_s/2^{l_2} + C' * q_s^{s'} + q_s/2^{l_1}. \quad (8)$$

$\Phi_6$ : in this game, the hash oracle  $H_1$  is replaced by the pri-

vate hash oracle  $H_1'$  to calculate the session key.  $H_1'$  is unavailable to  $\mathcal{A}$ . Hence,

$$\Pr[\chi_6] = \frac{1}{2}. \quad (9)$$

$\Phi_6$  has no difference with  $\Phi_5$ , unless  $\mathcal{A}$  has asked Hash query  $H_1(r_3N_i\|D_i)$ . This event is denoted by  $\Gamma_1$ . Hence,

$$|\Pr[\chi_6] - \Pr[\chi_5]| \leq \Pr[\Gamma_1]. \quad (10)$$

If  $\mathcal{A}$  has asked Hash query  $H_1(r_3N_i\|D_i)$ , when picking an item from  $\Lambda_H$ , we can get a solution of ECDHP with probability  $1/q_h$ . Hence,

$$\Pr[\Gamma_1] \leq q_h \text{Adv}_P^{\text{ECDHP}}. \quad (11)$$

From (3)–(11), we have

$$\begin{aligned} \text{Adv}_P^{\text{ake}}(\mathcal{A}) \leq & 2C' * q_s^{s'} + \frac{(q_s + q_e)^2}{p} + \frac{6q_s + q_h^2}{2^{l_1}} \\ & + \frac{2q_s + q_b^2}{2^{l_2}} + 2q_h \text{Adv}_P^{\text{ECDHP}}. \end{aligned} \quad (12)$$

**5.3. Further Security Analysis.** This section demonstrates that the proposed scheme is immune to known attacks and provides various desirable security properties.

**5.3.1. Mutual Authentication.** In our scheme, the cloud server authenticates the user by checking if  $D_iP = A_i + C_i \cdot \text{PUB}_i$ .  $D_i$  is a signature calculated based on user private key  $d_i$ . Only the user  $U_i$  who has the private key  $d_i$  can calculate a valid  $D_i$ . In addition, the user validates the cloud server by checking if  $L_i^* = L_i$ . Actually, the user authenticates the cloud server based on  $B_i = r_1 \text{PUB}_j = k_j A_i$ . In the login request,  $D_i$  is encrypted under the key  $B_i$ . Except the user  $U_i$ , only the cloud server  $CS_j$  who has the secret key  $k_j$  can compute  $B_i$  and retrieve  $D_i$  from  $E_i$  and generate a valid authenticate value  $L_i$ .

**5.3.2. Session Key Agreement.** The user and the cloud server generate a session key  $\text{SK} = H_1(r_3r_2P\|D_i)$ . The session key is composed of  $r_3r_2P$  and  $D_i$ .  $r_3r_2P$  is generated using elliptic curve Diffie-Hellman key exchange, and it guarantees forward secrecy.  $D_i$  is generated based on user's private key, and it guarantees the resistance of session-specific temporary information attack.

**5.3.3. User Anonymity.** In our scheme, the user identity is encrypted under the key  $B_i$ . As ECDHP is intractable, only the user who knows the random number  $r_1$  and the cloud server who has the secret key  $k_j$  can retrieve  $\text{ID}_i$  from  $E_i$ . Additionally, the random numbers  $r_1$  and  $r_2$  are involved in the login request  $\{A_i, E_i\}$ . The login requests are different in each session. Thus, the proposed scheme preserves user untraceability.

**5.3.4. Offline RC.** In the authentication phase, the user and the cloud server can perform mutual authentication and

TABLE 3: Security features Comparisons.

Security properties	Tsai and Lo [3]	He et al. [26]	Irshad et al. [27]	Mo et al. [28]	Our scheme
User anonymity	×	√	√	√	√
Resist server impersonation attack	×	√	×	√	√
Resist offline guessing attack	√	√	√	√	√
Resist stolen-verifier attack	√	√	√	×	√
Resist denial of service attack	√	√	√	×	√
Resist replay attack	√	√	√	×	√
Resist known session-specific temporary information attack	×	×	×	×	√
Forward secrecy	√	√	√	√	√
Three-factor secrecy	–	–	√	–	√
Efficiency for wrong password and biometric detection	×	×	√	√	√
Offline RC	√	√	√	–	√
Single/multi server	Multiserver	Multiserver	Multiserver	Single-server	Multiserver
Cryptography primitives	Paring	Paring	Paring	ECC	ECC

session key agreement without the aid of RC. It reduces the number of interacted messages. Correspondingly, it helps to reduce communication and computing overheads.

**5.3.5. Forward Secrecy.** The session key is computed based on  $SK = H_1(r_3r_2P \| D_i \| r_3, PUB_i)$ .  $r_3r_2P$  is generated using Diffie-Hellman key exchange. Due to the intractability of ECDHP, even the attacker obtains the long-term secret, he is unable to retrieve  $r_3r_2P$  from  $F_i$  and  $N_i$ . The proposed scheme preserves forward secrecy.

**5.3.6. Resist Session-Specific Temporary Information Attack.** Suppose that the random numbers  $r_2$  is compromised. The adversary computes  $r_3F_i$ . However, as  $B_i$  is unavailable, the adversary cannot obtain  $D_i$ .

Suppose that the random number  $r_3$  is compromised. The adversary cannot obtain  $N_i$  and  $D_i$ , as  $B_i$  is unavailable. The adversary can neither obtain  $D_i$  or  $r_3N_i$ .

As a result, the adversary cannot reveal the session key when the random number is compromised.

**5.3.7. Resist Forgery Attack.** In our scheme, the user computes the signature  $D_i$  based on the private key  $d_i$  to authenticate the message  $\{A_i, E_i, N_i\}$ . Afterwards, the cloud server uses the shared session key SK to authenticate the message  $\{F_i, L_i\}$ . Finally, the user uses the shared session key SK to authenticate the message  $\{M_i\}$ . As the secret key  $d_i$  and SK are unavailable, the adversary cannot produce a valid message.

**5.3.8. Resist Replay Attack.** In the proposed scheme, the cloud server authenticates the user by checking the validity of the messages  $\{A_i, E_i, N_i\}$  and  $\{M_i\}$ . If the adversary replays  $\{A_i, E_i, N_i\}$ , as he cannot produce a valid  $\{M_i\}$ , ultimately, the authentication fails. If the adversary replays  $\{F_i, L_i\}$  and  $\{M_i\}$ , as the random numbers selected in each session are different, the authentication fails. Hence, the proposed scheme can resist replay attack.

**5.3.9. Resist Insider Attack.** The user cannot impersonate the cloud server without cloud server's private key. Similarly, the

TABLE 4: Executing time of some cryptography operations.

Cryptography operations	Symbols	Running time (ms)	
		User	Server
Map-to-point hash function	$T_{PH}$	33.582	5.493
Bilinear paring	$T_B$	32.713	5.427
Elliptic curve point multiplication	$T_P$	13.405	2.165
Elliptic curve point addition	$T_A$	0.081	0.013
Exponentiation operation	$T_E$	2.249	0.339
Hash function	$T_H$	0.056	0.007

cloud server cannot impersonate the user without user's private key. The other users cannot pretend to be the user  $U_i$ , as he cannot generate a valid signature of  $U_i$ . The other cloud servers cannot pretend to be the cloud server  $CS_j$ , as he cannot decrypt  $E_i$  to get  $D_i$ . Our scheme is resistance to insider attack.

**5.3.10. User Friendliness.** The proposed scheme provides user friendliness. Firstly, the proposed scheme adopts multiserver architecture. The user only needs to register once to access multiple servers. Secondly, in the authentication phase, the registration center is offline, and the user can access the cloud server directly without interacting with the registration center. Thirdly, the proposed scheme supports smartcard revocation, efficiency for wrong password and biometric detection, and password and biometric update.

**5.3.11. Three-Factor Secrecy.** The fuzzy verification  $Z_i$  makes our scheme that is immune to offline guessing attack. Even if the adversary compromises two kinds of authentication factors, the other one is still unavailable. In addition, for the adversary, the only way to retrieve  $d_i$  is to break the password, the biometric, and the smart card at the same time. Without  $d_i$ , the adversary cannot impersonate the user. Hence, the proposed scheme preserves three-factor secrecy.

TABLE 5: Computation costs of related schemes.

Computation cost	User (ms)	Server (ms)	Total (ms)
Tsai and Lo [3]	$T_{PH} + 4T_P + 2T_A + T_E + 5T_H$ (89.893)	$2T_B + 2T_P + 2T_A + 2T_E + 4T_H$ (16.096)	105.989
He et al. [26]	$T_{PH} + 3T_P + 2T_E + 4T_H$ (78.519)	$2T_P + 2T_A + 2T_E + 5T_H$ (11.773)	90.292
Irshad et al. [27]	$1T_B + 4T_P$ (86.333)	$2T_B + 3T_P$ (17.349)	103.682
Mo et al. [28]	$3T_P + T_A + 6T_H$ (40.632)	$3T_P + T_A + 7T_H$ (6.557)	47.189
Our scheme	$4T_P + 6T_H$ (53.956)	$5T_P + 4T_H$ (10.853)	64.809

TABLE 6: Communication costs of related schemes.

	Tsai and Lo [3]	He et al. [26]	Irshad et al. [27]	Mo et al. [28]	Our scheme
Communication cost	4320 bits	3296 bits	4288 bits	2720 bits	3584 bits

## 6. Performance Comparisons

The comparative analysis of our scheme and the relevant schemes [3, 26–28] is presented in this section. Our scheme and the relevant schemes are evaluated from two aspects, i.e., security properties and computation and communication overheads.

Table 3 presents the security analysis results of relevant schemes. The security attributes include user anonymity and three-factor secrecy, as well as the resistance of usual attacks. Besides, the characteristics of the proposed schemes and relevant schemes are also detailed in Table 3. The relevant schemes [3, 26, 27] adopt multiserver architecture, and RC is offline in the authentication phase, while Mo et al.'s scheme adopts single-server architecture. Tsai et al.'s scheme, He et al.'s scheme, and Irshad et al.'s scheme are bilinear paring-based schemes, while Mo et al.'s scheme and our scheme are ECC-based schemes. From Table 3, we witness that the relevant schemes have more or less weaknesses, while the proposed scheme can remedy the security defects of relevant schemes and provides desirable security properties. It shows that the proposed scheme has better security than the relevant schemes.

In accordance with [26], the user uses a mobile device to access the cloud server, the cloud server is deployed in a personal computer, and the executing time of relevant cryptography operations is presented in Table 4. The computation costs of our scheme and the relevant schemes are evaluated as shown in Table 5. The running time of the proposed scheme is 80.379 ms. The running time of the relevant schemes [3, 26–28] is 105.989 ms, 90.292 ms, 103.682 ms, and 47.189 ms, respectively.

To evaluate the communication cost, we suppose that the user identity is 32 bits, the point on the elliptic curve group is 1024 bits, and the hash value is 160 bits. The login request query in [3, 26, 27] is 32 bits. As shown in Table 6, the communication cost of the proposed scheme is 3584 bits. The communication costs of the relevant schemes [3, 26–28] are 4320 bits, 3296 bits, 4288 bits, and 2720 bits, respectively.

Figure 5 presents the comparison of total computation costs, the computation costs of user end, and the computation costs of cloud server. Figure 6 presents the communication cost comparison. In terms of the communication cost,

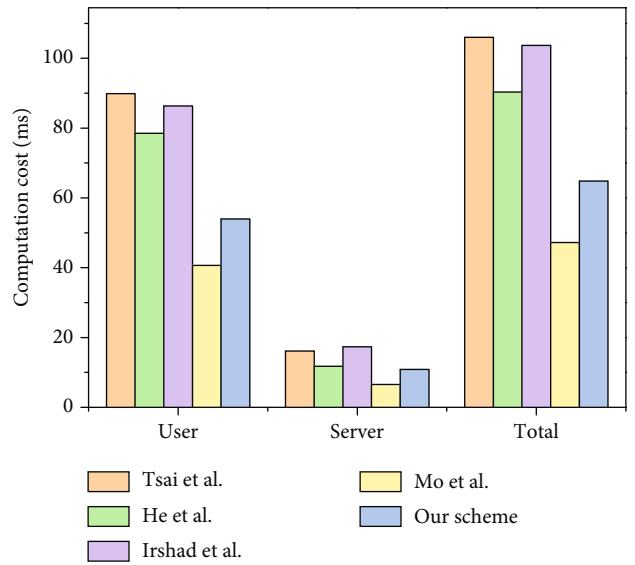


FIGURE 5: Computation cost comparisons.

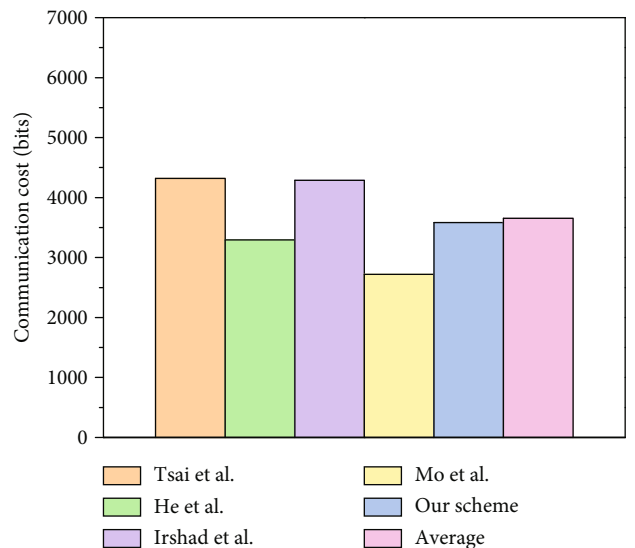


FIGURE 6: Communication cost comparison.



our scheme is in third place and better than the average communication cost. In terms of the total computation cost, user's computation cost, and server's computation cost, the proposed scheme is second only to Mo et al.'s scheme. However, Mo et al.'s scheme has limitations like stolen-verifier attack and denial of service attack; particularly, its single-server architecture is not applicable to the mobile cloud computing environment.

In a nutshell, our scheme provides more security attributes and has low computation and communication costs. Among the relevant schemes, the security features of He et al.'s scheme are the closest to our scheme. However, the computation cost of our scheme is 0.72 times of He et al.'s scheme. Our scheme achieves balanced security and efficiency. Compared with the relevant schemes, our scheme is more applicable to mobile cloud computing.

## 7. Conclusion

In this paper, we demonstrate that Mo et al.'s scheme has limitations such as stolen-verifier attack and denial of service attack. Most notably, its single-server architecture is not applicable to MCC. To enhance the security, we present a provably secure ECC-based three-factor authentication scheme. Security analysis shows that our scheme is immune to known attacks and provides user friendliness. Performance comparisons indicate that our scheme provides more security attributes and incurs low computation and communication cost. Our scheme is more applicable to MCC. As post-quantum security has become the focus issue of researchers, we plan to use lattice-based key exchange [36] and smooth projective hash functions [37] to construct a quantum-resistant scheme at the next step.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare no conflict of interest.

## Acknowledgments

This research was funded by the National Key Research and Development Program of China (No. 2018YFB0803600 and No. 2017YFB0801903) and by the National Natural Science Foundation of China (No. 61831003 and No. 61897069).

## References

- [1] A. Ghose, A. Goldfarb, and S. P. Han, *How is the mobile internet different?*, vol. 24, no. 3, 2012 Social Science Electronic Publishing, 2012.
- [2] P. Gope and A. K. Das, "Robust anonymous mutual authentication scheme for  $n$ -Times ubiquitous mobile cloud computing services," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1764–1772, 2017.
- [3] J. L. Tsai and N. W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, vol. 9, no. 3, pp. 805–815, 2015.
- [4] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.
- [5] E. J. Yoon, K. Y. Yoo, C. Kim, Y. S. Hong, M. Jo, and H. H. Chen, "A secure and efficient sip authentication scheme for converged VOIP networks," *Computer Communications*, vol. 33, no. 14, pp. 1674–1681, 2010.
- [6] F. Wang, G. Xu, and L. Gu, "A secure and efficient ECC-based anonymous authentication protocol," *Security and Communication Networks*, vol. 2019, Article ID 4656281, 13 pages, 2019.
- [7] F. Wei, P. Vijayakumar, Q. Jiang, and R. Zhang, "A mobile intelligent terminal based anonymous authenticated key exchange protocol for roaming service in global mobility networks," *IEEE Transactions on Sustainable Computing*, vol. 99, pp. 2377–3782, 2018.
- [8] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.
- [9] X. Li, W. Qiu, D. Zheng, K. Chen, and J. Li, "Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 2, pp. 793–800, 2010.
- [10] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, "Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1382–1392, 2017.
- [11] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2018.
- [12] D. Wang and P. Wang, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks," *Ad Hoc Networks*, vol. 20, pp. 1–15, 2014.
- [13] C. Wang, K. Ding, B. Li et al., "An enhanced user authentication protocol based on elliptic curve cryptosystem in cloud computing environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3048697, 13 pages, 2018.
- [14] D. Wang, D. He, P. Wang, and C. Chu, "Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2015.
- [15] S. Kumari, X. Li, F. Wu, A. K. Das, K. R. Choo, and J. Shen, "Design of a provably secure biometrics-based multi-cloud-server authentication scheme," *Future Generation Computer Systems*, vol. 68, pp. 320–330, 2017.
- [16] F. Wang, G. Xu, C. Wang, and J. Peng, "A provably secure biometrics-based authentication scheme for multiserver environment," *Security and Communication Networks*, vol. 2019, Article ID 2838615, 15 pages, 2019.
- [17] L. H. Li, L. C. Lin, and M. S. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.
- [18] C. Wang, G. Xu, and J. Sun, "An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for

- wireless sensor networks,” *Sensors*, vol. 17, no. 12, p. 2946, 2017.
- [19] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, “Security and privacy challenges in mobile cloud computing: survey and way ahead,” *Journal of Network and Computer Applications*, vol. 84, pp. 38–54, 2017.
- [20] V. Odelu, A. K. Das, S. Kumari, X. Huang, and M. Wazid, “Provably secure authenticated key agreement scheme for distributed mobile cloud computing services,” *Future Generation Computer Systems*, vol. 68, pp. 74–88, 2017.
- [21] L. Xiong, D. Peng, T. Peng, and H. Liang, “An enhanced privacy-aware authentication scheme for distributed mobile cloud computing services,” *KSII Transactions on Internet and Information Systems*, vol. 11, no. 12, pp. 6169–6187, 2017.
- [22] Q. Jiang, J. Ma, and F. Wei, “On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services,” *IEEE Systems Journal*, vol. 12, no. 2, pp. 2039–2042, 2018.
- [23] Q. Feng, D. He, S. Zeadally, and H. Wang, “Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment,” *Future Generation Computer Systems*, vol. 84, pp. 239–251, 2017.
- [24] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, “A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment,” *Future Generation Computer Systems*, vol. 78, pp. 1005–1019, 2018.
- [25] P. Wang, B. Li, H. Shi, Y. Shen, and D. Wang, “Revisiting anonymous two-factor authentication schemes for IoT-enabled devices in cloud computing environments,” *Security and Communication Networks*, vol. 2019, Article ID 2516963, 13 pages, 2019.
- [26] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, “Efficient privacy-aware authentication scheme for mobile cloud computing services,” *IEEE Systems Journal*, vol. 12, no. 2, pp. 1621–1631, 2018.
- [27] A. Irshad, S. A. Chaudhry, M. Shafiq, M. Usman, M. Asif, and A. Ghani, “A provable and secure mobile user authentication scheme for mobile cloud computing services,” *International Journal of Communication Systems*, vol. 32, no. 14, article e3980, 2019.
- [28] J. Mo, Z. Hu, H. Chen, and W. Shen, “An efficient and provably secure anonymous user authentication and key agreement for mobile cloud computing,” *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 4520685, 12 pages, 2019.
- [29] Z. Li, D. Wang, and E. Morais, “Quantum-safe round-optimal password authentication for mobile devices,” *IEEE Transactions on Dependable and Secure Computing*, 2020.
- [30] N. Koblitz, “Elliptic curve cryptosystems,” *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.
- [31] D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
- [32] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication,” *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.
- [33] F. Wang, G. Xu, G. Xu, Y. Wang, and J. Peng, “A robust IoT-based three-factor authentication scheme for cloud computing resistant to session key exposure,” *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 3805058, 15 pages, 2020.
- [34] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, “Zipf’s law in passwords,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.
- [35] D. Wang and P. Wang, “On the implications of Zipf’s law in passwords,” in *Proc. Eur. Symp. Res. Comput. Secur.*, pp. 111–131, Cham, 2016.
- [36] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, “Post-quantum key exchange—a new hope,” in *Proc. USENIX SEC 2016*, pp. 327–343, Austin, TX, 2016.
- [37] J. Katz and V. Vaikuntanathan, “Smooth projective hashing and password-based authenticated key exchange from lattices,” in *Proc. ASIACRYPT 2009*, pp. 636–652, Berlin, Heidelberg, 2009.

## Review Article

# Security Analysis of Out-of-Band Device Pairing Protocols: A Survey

Sameh Khalfaoui <sup>1,2</sup>, Jean Leneutre <sup>1</sup>, Arthur Villard,<sup>2</sup> Jingxuan Ma,<sup>2</sup> and Pascal Urien <sup>1</sup>

<sup>1</sup>LTCL, Télécom Paris, Institut Polytechnique de Paris, France

<sup>2</sup>EDF R&D, France

Correspondence should be addressed to Sameh Khalfaoui; sameh.khalfaoui@edf.fr

Received 5 August 2020; Revised 28 October 2020; Accepted 24 November 2020; Published 30 January 2021

Academic Editor: Qi Jiang

Copyright © 2021 Sameh Khalfaoui et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Numerous secure device pairing (SDP) protocols have been proposed to establish a secure communication between unidentified IoT devices that have no preshared security parameters due to the scalability requirements imposed by the ubiquitous nature of the IoT devices. In order to provide the most user-friendly IoT services, the usability assessment has become the main requirement. Thus, the complete security analysis has been replaced by a sketch of a proof to partially validate the robustness of the proposal. The few existing formal or computational security verifications on the SDP schemes have been conducted based on the assessment of a wide variety of uniquely defined security properties. Therefore, the security comparison between these protocols is not feasible and there is a lack of a unified security analysis framework to assess these pairing techniques. In this paper, we survey a selection of secure device pairing proposals that have been formally or computationally verified. We present a systematic description of the protocol assumptions, the adopted verification model, and an assessment of the verification results. In addition, we normalize the used taxonomy in order to enhance the understanding of these security validations. Furthermore, we refine the adversary capabilities on the out-of-band channel by redefining the replay capability and by introducing a new notion of delay that is dependent on the protocol structure that is more adequate for the ad hoc pairing context. Also, we propose a classification of a number of out-of-band channels based on their security properties and under our refined adversary model. Our work motivates the future SDP protocol designer to conduct a formal or a computational security assessment to allow the comparability between these pairing techniques. Furthermore, it provides a realistic abstraction of the adversary capabilities on the out-of-band channel which improves the modeling of their security characteristics in the protocol verification tools.

## 1. Introduction

With the growing demand for IoT objects for both the personal and the industrial contexts, the use of a decentralized device-to-device (D2D) communication system has become a necessity for numerous applications in the context of Internet of Things (IoT). This decision is based on the inefficiency of a centralized communication solution to meet the scalability and the interoperability goals. Therefore, the protection of this communication channel requires the use of a secure key establishment protocol between the devices, known as *secure device pairing* (SDP). This process ensures that the commu-

nicating nodes agree on the same symmetric encryption key, which represents an initial trust establishment between devices that have no preshared knowledge (a certificate, a shared password, or a symmetric key). The no prior secret condition is motivated by two reasons: the unfeasibility of exploiting a public key infrastructure (PKI) due to the growing numbers of heterogeneous IoT devices, and the *zero-trust* policy that disapproves of trusting the manufacturer with delivering the initial preshared pairing keys to avoid any vulnerabilities or breaches related to a third party.

Numerous secure device pairing solutions have been proposed to securely establish a shared key between a number of

devices that do not share any prior security knowledge. These techniques can be divided into two main categories. The first one ensures the confidentiality and the data authentication of the key through a proof of copresence based on the randomness of the ambient environment and it is better known as context-based pairing or zero-interaction protocols (ZIP) [1, 2]. The second technique relies on an auxiliary channel with specific security properties to send an information that validates what has been exchanged on the main insecure channel, referred to as the in-band channel. However, in this state of art, we will only discuss the security analysis of the out-of-band secure device pairing schemes that rely on an auxiliary channel [3–5].

The use of the secondary channel is due to the unfeasibility of performing the authentication based on a single channel that is controlled by a Dolev-Yao intruder [6], as demonstrated in [7] using BAN logic analysis [8]. This powerful adversary is assumed to have a perfect knowledge of the protocol and he is able to overhear, block, delay, replay, and forge any transmission over that channel. However, he is not able to perform any computational attacks against the cryptographic functions. As a consequence of adopting this intruder model, the usage of the main insecure channel without having preshared secrets is not sufficient to provide the desired security guarantees for the key exchange process. Therefore, there is a need for an auxiliary communication link on which the authentication of the exchanged keys can happen. These channels can be constructed based on audio, visual, or haptic transmissions. Due to their special nature and their communication properties, they provide an initial level of security that is sufficient to primarily guarantee the integrity, the data origin authenticity, and the demonstrative identification [9], which is ensuring that the communicating devices on these channels are the intended ones for pairing. Other security objectives might be provided in some cases such as the confidentiality. These assumptions on the OoB channel reduce the attacker capabilities in comparison with his abilities on the main insecure channel. On the other hand, there is another variant of secure device pairing schemes that uses the randomness of the ambient environment in order to securely establish a shared key between the intended devices. These protocols might rely on external factors with respect to the human user such as the radio environment [10–12], the acoustic surroundings [13, 14], or other random physical patterns [15–18]. However, numerous context-based pairing research works in the field of wireless body area network (WBAN) rely on specific human-centric biometrics that are extracted by the sensors attached to the user which is more suitable for the implantable medical devices (IMD) [19–22]. These collected random features are used as the secure element in the protocol execution. Nonetheless, the evaluation of these contextual pairing schemes is considered out of the scope of this work. Readers eager to learn more about these protocols and their applications can consult these review articles [1, 2, 23].

In the literature, a variety of surveys [3–5] have addressed the out-of-band pairing problem from the security perspective. In the work of Nguyen and Roscoe [5], the authors conducted a study on the authentication process involving secure

device pairing schemes that rely on a manual transfer of a short authentication string (SAS). They discussed the costs related to the cryptographic techniques applied in the protocol design and the required communication between the pairing participants. However, this work proposed a classification of the out-of-band channels according to some assumptions about their threat models which appeared to us to be unrealistic in some cases such as the feasibility of a delay attack. In the work of Mirzadeh et al. [4], the authors extended the work of Nguyen and Roscoe [5] by conducting an extensive survey on a number of pairwise and groupwise device pairing protocols based on a similar classification of the out-of-band channels. Although the work tends to mention the results of the conducted formal or the computational security proofs, it does not describe the evaluated properties nor discuss their associated assumptions, and as a consequence, it does not offer a complete basis to compare the provided security of the different protocol. In addition, a great body of work on SDP tends to investigate the same authentication and confidentiality properties under different definitions that drift away from the commonly known specifications such as the ones given in the work of Lowe [24]. Therefore, these verification results are difficult to interpret. Furthermore, the security analysis using the protocol verification tools has not been discussed even though multiple research works [25–27] have adopted these formal methods to evaluate the security of their proposals based on a predefined set of authentication properties. Also, we have noticed that numerous SDP schemes are based on a threat model, inspired from the Dolev-Yao intruder capabilities [6]. This model allows the adversary to replay messages on the out-of-band channel while guaranteeing the integrity of the exchanged information. On the other hand, the act of forging a message that pleases the attacker is deemed unfeasible and will be, somehow, detected. These two assumptions might be plausible when the two devices have a preshared secret that is used to sign the OoB messages which force the attacker to only replay previous exchanges. Unfortunately, this is not the case for the ad hoc secure device pairing due to the lack of preshared security knowledge between the pairing participants. Thus, it makes these assumptions not valid and it might lead to vulnerabilities when the scheme is deployed. Furthermore, the previously described intruder model assumes that the attacker is able to delay any out-of-band transmission for a desired given time. In the context of a direct communication channel, this specific action is highly dependent on the feasibility of blocking a message and replaying it afterwards. Therefore, if the replay attack is not considered feasible, then the delay assumption is no longer valid. In the work of Fomichev et al. [3], the authors have provided a systematic modeling of the pairwise pairing procedure by describing its three main components: the out-of-band channel, the user involvement, and the pairing context. Also, they outlined the characteristics of the OoB channels by detailing their communication properties, by summarizing some of their known vulnerabilities and by identifying some of their main usability advantages in the IoT context. However, their analysis does not give a detailed security assessment of the SDP schemes. The focus in their

analysis of the protocols is more oriented toward the usability aspects than the security. Even though our main focus is related to the formal or computational security assessment of numerous SDP schemes, we point out the importance of enhancing the usability of these pairing processes in order to facilitate their ease-of-adoption. Readers eager to learn more about the usability and the human-in-the-loop aspect in the secure device pairing procedure can consult these review papers [3, 28–30].

In this work, we focus on providing a comprehensive study on the existing formal and computational security proofs that are conducted on a selection of secure device pairing schemes. This review clearly lays out the definitions of the chosen security properties, the adopted verification model, the associated protocol assumptions, and an assessment of the verification results. Although every analysis tends to use its own terminologies and its own definitions, we normalize the used taxonomy in order to enhance the understanding of these security validations. Also, we refine the adversary model that has been adopted by multiple pairing proposals by eliminating the replay capability and by introducing a new notion of delay that is based on the protocol structure rather than the out-of-band channel characteristics. These modifications are motivated by the urge to have a security model that is adequate to the ad hoc device pairing context and assumptions in order to facilitate their validation and deployment in a realistic scenario. Based on our security model, we classified a selection of out-of-band channels based on an evaluation of their achieved security goals. In addition, we describe an advanced threat model that consists of violating two security guarantees: *the demonstrative identification* and *the device integrity* (the latter property outlines that one of the pairing participants is under the control of the adversary). This adversary model has yielded a recently published attack, called *misbinding* [27], that targets the majority of the device pairing schemes.

This work is aimed at introducing and motivating the use of the formal and the computational security analysis in the process of validating the robustness of the secure device pairing schemes. Also, it serves as a road map for properly designing an SDP protocol that achieves the desired security goals and that can be applicable to realistic scenarios by providing the adequate criteria for choosing the appropriate out-of-band channel. In addition, it sheds light on the recently discovered attacks and vulnerabilities that affect the robustness of the SDP protocols.

The main contributions of this paper are summarized as follows:

- (i) We conduct a comprehensive study on the existing formal and computational security proofs that evaluate a selection of secure device pairing schemes relying on an out-of-band channel
- (ii) We enhance the threat model, adopted by numerous SDP proposals to describe the attacker action on the OoB channel, by eliminating the replay assumption and by introducing a new realistic approach to the delay attack based on the structure of the protocol.

Then, we derive six categories of the out-of-band channels based on their achieved security goals in our threat model

- (iii) We conduct a classification of a commonly used OoB channels based on the security categories derived previously
- (iv) We discuss the recently published misbinding attack by explaining its origin, the adopted adversary model, and some of the proposed mitigations
- (v) We provide a number of secure pairing design recommendations for future SDP designers and we highlight a number of future challenges, based on identified security weaknesses, where SDP research is demanded

The rest of the paper is organized as follows. Section 2 focuses on the out-of-band channels by describing the limitations of the widely adopted OoB adversary model and it presents our enhancement proposals with respect to the attacker capabilities and the security guarantees that should be evaluated. Also, it discusses the security and the usability properties of a selection of the commonly used out-of-band channels. In addition, it provides a classification based on our refined threat model. Section 3 describes a number of SDP schemes that have been either formally or computationally verified. Thus, other SDP proposals with only a sketch of a security proof are considered out of the scope of this work. Furthermore, it discusses an advanced threat model that assumes that one of the pairing participants is compromised and that the user unintentionally initiates the pairing with a malicious device. These assumptions have been demonstrated feasible and they lead to a misbinding attack that falsely establishes the pairing with a distant malicious object. Also, it focuses on a number of common vulnerabilities and security considerations when designing a pairing protocol that is based on an out-of-band channel. Section 4 highlights four main aspects: the most common design vulnerabilities in the out-of-band pairing protocols, the recommendations of the necessary mitigations, a description of the limitations of the security analysis conducted on the SDP schemes, and the future areas that need to be further studied regarding this matter. Lastly, Section 5 concludes our work.

## 2. Out-of-Band Channel Overview

**2.1. Refined Out-of-Band Threat Model.** In this study, we adopt the Dolev-Yao intruder model [6] on the in-band channel where he has complete control over the network. We assume that the attacker is able to perform the following actions: overhear, block, delay, replay, and forge any message on the channel. This latter action includes a modification attempt on a previously captured legitimate message. Due to the absence of any preestablished security information, the attacker has the same level of knowledge as the legitimate devices which eliminate any possibility of performing a secure key establishment using only the in-band channel, as proved in [7] using BAN logic analysis [8].

This is obviously not the case for the out-of-band channel since it is assumed by design to be partially out of reach of the adversary. Therefore, it should guarantee at least the integrity and the data origin authenticity of the messages. Also, the confidentiality property on the OoB channel, referred to as *private OoB* [4], is demanded by some SDP schemes ([31, 32]). This assumption is hard to obtain and might ultimately lead to vulnerabilities in the protocol design [9]. The OoB channels reduce the attacker capabilities to overhearing, blocking, and delaying the authentication strings. Thus, the adversary cannot replay or forge a message without being exposed. These restrictions result in an authenticated out-of-band channel that is referred to as *public OoB* [4]. In some cases, the attacker might be given the capability to replay previously sent messages on the out-of-band channel and it is referred to as *weak OoB* [4].

Unfortunately, under the assumption that we have no prior security knowledge between the legitimate devices and the assumption that the attacker has perfect knowledge of the protocol execution, it is not realistic to assume that an adversary is only able to replay a message without having the power to forge a suitable one and send it on the peer-to-peer out-of-band channel, as adopted in a great body of research work. We state that, based on this logic, any SDP scheme that allows an adversary to replay but not to inject their own messages under the assumption that we have no preshared secret is ultimately vulnerable. Therefore, while considering the presence of a vigilant user, we will model our attacker capabilities by only three actions: overhear, block, and inject any exchange on the OoB channel. The latter action includes the transmission of either a previously captured or a freshly constructed message. Also, the delay capability can be hard to achieve directly over the peer-to-peer out-of-band channel without considering the combination of the block and the replay actions. However, it can be considered possible using the attacker capability to perform this action on a previous exchange over the in-band channel that was intended to trigger the OoB transmission. In this case, the act of delaying the previous insecure exchange will result in stopping the protocol execution for the same amount of time which, consequently, will lead to a delay over the reception of the OoB transmission. Therefore, this action targets the protocol execution in order to affect the out-of-band channel which affects any protocol that has an in-band exchange prior to the OoB transmission. As an example of a protocol structure that is immune against this malicious act, the well-known device pairing scheme, *talking to strangers* [9], starts by a bidirectional OoB exchange of the public key hashes which, according to our model, does not grant the adversary the power to perform a delay attack. In order to target all the cases, we consider the delay as an action that is dependent on the protocol structure instead of the OoB channel specifications.

These previously described actions are assessed to evaluate the following security objectives on the out-of-band channel that we deem necessary to guarantee the required security of the OoB exchange under our adversary model:

- (i) Confidentiality (C) [33]: the information, sent over the channel, can only be accessed by the authorized

pairing parties. Therefore, the attacker cannot overhear the communication

- (ii) Data freshness (DF) [33]: the information, sent over the channel, cannot be replayed by a malicious actor. Therefore, the attacker cannot inject any old messages on the channel
- (iii) Data origin authentication (DOA): any receiver of the information, transmitted on the channel, is able to authenticate its sender. Therefore, the attacker cannot inject his own messages on the channel as if they were coming from a legitimate sender
- (iv) Liveness (L) [34]: any information, transmitted over the channel, is eventually received by the intended party. Therefore, the attacker cannot block any transmission over the channel
- (v) Channel availability (CA): any information, transmitted over the channel, is received at the intended protocol execution order. Therefore, the attacker cannot delay any transmission over the channel

Based on these five security goals, we can conduct a more refined and realistic out-of-band channel classification. We will have six main channel types:

- (i) Confidential OoB: all the security goals are guaranteed. Therefore, the adversary has no capabilities
- (ii) Delayable-confidential OoB: only the channel availability assumption is not guaranteed. Therefore, the adversary can only delay the transmission
- (iii) Protected OoB: only the confidentiality goal does not hold. This means that the attacker is only capable of overhearing the communication
- (iv) Delayable-protected OoB: only the confidentiality and the channel availability goals do not hold. This means that the attacker is only capable of overhearing and delaying the communication
- (v) Authentic OoB: only the integrity, the data freshness, the data origin authentication, and the channel availability goals are achieved. Therefore, the adversary is capable of blocking and overhearing the OoB channel
- (vi) Delayable-authentic OoB: only the integrity, the data freshness, and the data origin authentication security goals are achieved. Therefore, the adversary is capable of blocking, delaying, and overhearing the OoB channel

The confidential channel represents the most secure channel since it achieves all the security goals desired. On the other hand, the delayable-authentic represents the minimum required OoB channel to ensure the security of the device pairing process, as shown in Table 1.

*2.2. Out-of-Band Security Classification.* The majority of the existing pairing solutions rely on an auxiliary channel with

TABLE 1: Attacker capabilities on the in-band and out-of-band channels.

Channel type	Adversary powers				Achieved security goals					
	Overhear	Block	Inject	Delay	Confidentiality	Integrity	Data freshness	Data origin authentication	Liveness	Channel availability
In-band channel	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
Confidential OoB	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Delayable-confidential OoB	✗	✗	✗	✓	✓	✓	✓	✓	✓	✗
Protected OoB	✓	✗	✗	✗	✗	✓	✓	✓	✓	✓
Delayable-protected OoB	✓	✗	✗	✓	✗	✓	✓	✓	✓	✗
Authentic OoB	✓	✓	✗	✗	✗	✓	✓	✓	✗	✓
Delayable-authentic OoB	✓	✓	✗	✓	✗	✓	✓	✓	✗	✗

specific security properties to send information that validates what has been exchanged on the in-band channel. The reason behind this diversity in the communication channel usage is that the authentication based on a single communication link is not feasible using BAN logic analysis [8]. As proven in [7], “Key-based device authentication between two previously unknown mobile devices in an ad-hoc computing environment is not possible using only a single wireless communication channel”. Therefore, using only the main insecure channel is not sufficient. Thus, there is a need for an auxiliary channel on which the authentication of the exchanged keys can happen. Known as *out-of-band* (OoB) channel, *location limited channel* (LLC), or *side channels* [9], these communication links can be constructed based on audio, visual, or haptic transmissions ([31, 35, 36]) and their goal is to guarantee the integrity of the transmitted information.

The major limitation of these channels is their low data rate which means that transferring long hashes or keys is not possible. In the work of Fomichev et al. [3], the described communication properties of the chosen out-of-band channels contradict the previous declaration. This fact is, simply, explained by the absence of the dedicated hardware on the commercial IoT devices due to cost optimization factors. Therefore, this constraint explains the long completion time of a 15-bit OoB exchange conducted in the work of Kumar et al. [30].

Some of the proposed schemes rely, more extensively, on the human user to interact with the devices and either *relay*, *compare*, or *generate* an information. These interactions make him the communication link itself known as human-computer interaction (HCI) channel [3]. The security objectives are assessed based upon the user behavior which makes them prone to human factor error that, if not well designed, might compromise the effective security of the protocol and its performance [29].

In this section, we will present both the security and the usability properties for a selection of the most common out-of-band channels based on our refined adversary model. Furthermore, we will be briefly introducing some of the existing schemes that take advantage of each of the selected OoB channels. Finally, the five security goals, defined in the adversary model in Subsection 2.1, will be used to classify these

chosen channels based on the security they offer while taking into account the presence of a vigilant user, as summarized in Table 2.

**2.2.1. Near-Field Communication (NFC).** NFC is a wireless communication technology used for point-to-point exchanges between two devices under the condition of *close physical proximity* as shown in Figure 1. These devices can be active or passive [37]. NFC chips are widely deployed and they are used in a wide variety of IoT devices.

(1) *Usability Properties.* As stated previously, NFC requires the two devices to be in a close proximity which means that the user is required to have a minimal intervention of putting the objects close to each other. The line of sight (LoS) transmission is not required which eliminates the need for a major user involvement in the case of aligning the two pairing parties. Due to its nonperceptibility property, this technology relies on the user vigilance to make sure that there is no suspicious behavior around them which is quite hard, especially for nonexpert users. This requirement represents a burden on the user and a drawback when it comes to the user friendliness aspect.

(2) *Security Properties.* The devices using NFC chips can be active in order to act as a contactless card reader or communicate with another object. It can also be passive in the case of a static message carrier such as a hash of a key or a password. This means that the risk of unauthorized readings can lead to a practical relay attack [38].

From a security perspective, the close proximity assumption plays a major role in protecting the devices from a sufficiently distant attacker since he is considered unable to overhear or interfere on the communication. Unfortunately, it has been proven possible in [39] where an eavesdropping attack on a commodity NFC-enabled mobile device has been successful from a distance up to 240 cm. Furthermore, a man-in-the-middle attack has been demonstrated in [40] between two NFC-enabled devices separated by a 10 cm distance. The fact that the security is provided based on a proximity assumption, an attacker can always violate such

TABLE 2: Channels classification based on the achieved security goals.

Out-of-band channel	Confidentiality	Integrity	Data freshness	Data origin authentication	Liveness	Channel classification
NFC	X	X	X	X	X	In-band
RFID	X	X	X	X	X	In-band
MM-waves	X	✓	✓	✓	X	Authentic
VC	X	✓	✓	✓	X	Authentic
Audio	X	✓	✓	✓	✓	Protected
Haptic	X	✓	✓	✓	✓	Protected

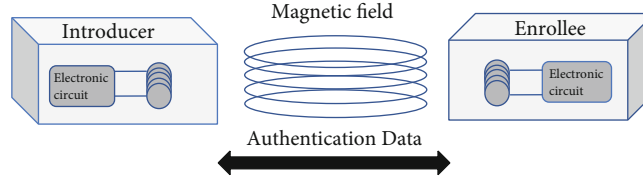


FIGURE 1: Communication model of a NFC technology.

requirement which does not make this out-of-band channel any better than the in-band channel because of its similar communication properties.

### (3) Proposed Schemes.

- (1) Push-button configuration (PBC) is part of the standardized Wi-Fi protected setup (WPS) [41] that introduces a pairing scheme using two options:
  - (i) Password token: the enrollee device will transmit a 32-byte random password to the NFC-enabled registrar. The same password will be used with the in-band registration protocol to provision the enrollee with WLAN configuration data
  - (ii) Connection handover: the two NFC-enabled devices exchange the hashes of their Diffie-Hellman public keys (exchanged previously on the in-band channel) using NFC to verify that they are communicating with the same device that was involved in the near-field communication
- (2) Secure Simple Pairing (SSP) is part of the standardized Bluetooth Secure Simple Pairing [42] that introduces a pairing scheme using an out-of-band option:
  - (i) Out of band: after the discovery phase via Bluetooth, the cryptographic authentication parameters as well as the identification information (Bluetooth device address) are sent over the OoB channel which has been reported to be resistant against MitM attacks

**2.2.2. Radio Frequency Identification Channel (RFID).** RFID is a wireless communication technology used for both indoor and outdoor identifications. These systems consist of small tags that emit stored identification information when inter-

rogated by an RFID reader which makes them a sort of an automatic identification system [43]. The majority of the used RFID tags are *passive* since they rely on the energy emitted by the RFID readers, as shown in Figure 2. We can find *active* tags having on-board their own power supply which makes them able to establish a bidirectional communication channel.

- (1) *Usability Properties.* This technology does not require any human intervention in the case of the high frequencies which make it more user-friendly and more appealing to nonexpert users. On the other hand, for the low frequencies, it has the same requirements as the NFC technology, described in Subsection 2.2.1.
- (2) *Security Properties.* For the low frequencies, RFID has similar security properties to the NFC technology stated in Subsection 2.2.1.

For the high frequencies, the range of the passive reads increases to reach 10 meters which makes an attacker able to retrieve the identification information and relay it since that kind of tags is very constrained and it responds to any reader [43]. Including the active tags and their long range (>100 m), this technology offers similar communication properties to what is used for the in-band channel. This makes the adversary in total control of the communication as stated in our adversary model in Subsection 2.1.

(3) *Proposed Schemes.* Noisy tag [45] is the injection of intentional noise, using an extra RFID tag (noisy tag), into an authentic channel making the eavesdropping process meaningless for the adversary. Only the legitimate reader (owner of the noisy tag) will be able to retrieve the original message from the noise-emitted signal. One downside to this scheme is that it does not protect the tag against an active attacker. It assumes that the active attacks require the adversary to be closer to the tag than in the case of eavesdropping and such active distance requirement can be circumvented by natural



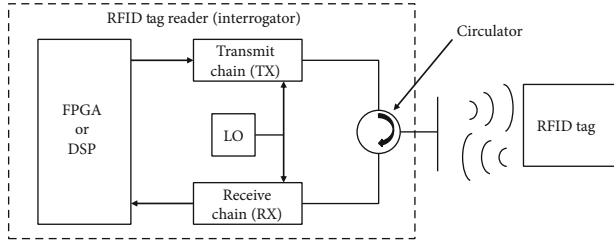


FIGURE 2: Block diagram of a RFID communication system [44].

barriers, e.g., in private areas (user surveillance, house, office, and building).

**2.2.3. Millimeter Waves (MM-Waves).** MM-waves is a wireless communication operating on the extremely high frequency (EHF) range. The high frequencies and their propagation properties make them useful for applications such as the transmission of large amount of data, cellular communications, and radar [46]. A standard IEEE 802.11ad [47] enables multigigabit wireless communications in the unlicensed 60 GHz band [48], as shown in Figure 3. This band is considered ideal for a variety of indoor applications since it supports data rates up to 7 Gbps [48].

*(1) Usability Properties.* The short-range requirement, similar to the NFC in Subsection 2.2.1, forces the user to be in close proximity of the two devices and to be vigilant of their surroundings in the covered area. Alongside with the penetration characteristic, the act of pairing devices from a distance is not feasible which is not convenient in the case of a smart-home containing multiple deployed IoT devices. As for the LoS condition, a user intervention during the pairing is crucial in order to set up the devices to face each other for a proper communication.

*(2) Security Properties.* The short-range penetration and LoS characteristics of the MM-waves provide a highly secure operation. This has been explained by the unfeasibility of a simple eavesdropping attack since the adversary has to be in the same room which would expose him to our vigilant user. However, as presented in [50], eavesdroppers can successfully intercept even highly directional transmissions using small-scale objects (from coffee cups to cell phones) as reflectors. These properties make the MitM attack hard for the attacker especially in a closed area where the walls create a natural barrier to the MM-wave emissions.

*(3) Proposed Schemes.* There are not many devices that support MM-waves, e.g., [51], but their popularity is on the rise. The previously described pairing scheme PBC from the standardized WPS [41] uses MM-waves as an out-of-band channel to perform the authentication process and it has been implemented on the HP advanced wireless dock (HP Elite x2 1011 G2 [52]). Even though the original version of the PBC scheme is vulnerable to MitM attacks, the close physical proximity, LoS, and no-penetration characteristics of the MM-waves force the attacker to be copresent which exposes him even by a benign user.

**2.2.4. Visible Communication (VC).** VC is a wireless communication technology that relies on modulating the visible spectrum using an illumination source such a display or an LEDs to transmit data. The short-range property of this technology is explained by the propagation distance of the emitting interface [53]. This technology includes multiple practices such as the use of a display-camera setup that shows a specific message (a QR code or a short authentication string) in order to create a short-range, interference-free out-of-band channel. The characteristics of the channel are directly dependent on the size of the screen to provide an independence of the view angle and the quality of the camera to guarantee a better detection, e.g., Pixnet [54]. However, this option assumes the existence of display and a camera on the transmitter and the receiver side which is not always the case for the low budget IoT devices. On the other hand, we can find the most common and most easily constructed variant that is referred to as visible light communication (VLC). A one-way VLC channel is described in Figure 4 as three main components: a transmitter, a channel, and a receiver.

*(1) Usability Properties.* Similar to the NFC in Subsection 2.2.1 and the millimeter waves in Subsection 2.2.3, the short-range requirement forces the user to be in close proximity of the two devices and to be vigilant of their surroundings in the covered area.

This monitoring act is more feasible from a user perspective since he is able to perceive any light emissions coming from an unauthorized source (potentially malicious).

Alongside with the penetration characteristic, the act of pairing devices from a distance is not feasible which is not convenient in the case of a smart home containing a wide variety of devices. As for the LoS condition, a user intervention during the pairing is crucial in order to set up the devices to face each other for proper communication.

The devices to be paired have to be equipped with at least a LED and a photosensor in the case of a unidirectional communication which is not the case for the constrained IoT products. On the other hand, the majority of devices are equipped with a display capable of performing the transmission but not a camera which means that the communication channel can only be unidirectional.

*(2) Security Properties.* Even though VLC might seem secure by design against eavesdropping especially when taking into account the LoS requirement and the no-penetration of solid objects such as the walls of the smart home, it has been proven in [55] that this attack is feasible and easy to perform through the door gaps, the keyholes, and the windows. These attack scenarios make use of the reflections of the light emissions and they provide low to no BER depending on the modulation scheme used by the transmitter.

Also an adversary can use a directional light to alter the transmitted message by sending pulses to the photosensor. This process is fairly easy to perform in an arbitrary way which means the attacker cannot predict the outcome of

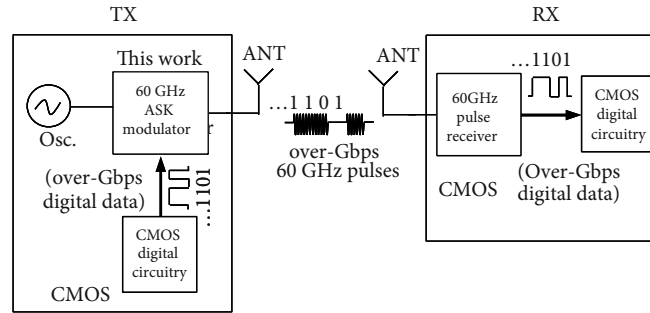


FIGURE 3: Block diagram of a millimeter wave communication system with a 60 GHz ASK (amplitude shift keying) modulator [49].

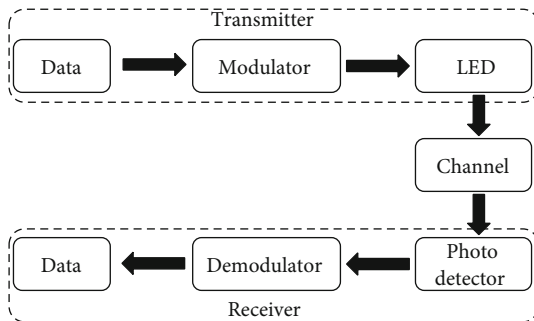


FIGURE 4: Block diagram of a VLC communication system [53].

the attack. Therefore, it will not be of a great impact on the pairing process and cause a MitM attack but it will lead to a denial of service (DoS). However, this technique might be useful to block the reception of the light pulse by saturating the photodetector on the receiving side.

One major threat when using a display-camera communication is the risk of replay attacks. This malicious act targets the liveness of the video captured by the camera. The attacker can easily record a previous conversation between a camera-enabled phone and an IoT object with a display using shoulder surfing or CCTVs [56]. Then, he replays the video to the camera in a way to pair with it. One solution to this issue is the analysis of this property by the comparison of the inertial measurements taken by the phone during the transmission and the motion analysis captured on the recorded video as better described in Figure 5 [57].

The data freshness property can be assured by the unfeasibility of any injection attacks on this out-of-band channel when the user vigilance assumption is assumed. In addition, the perceptibility of the light emissions and the LoS requirement facilitate the monitoring of the area surrounding the legitimate devices.

### (3) Proposed Schemes.

- (1) **Blinking light** [58]: after exchanging the key between the devices on the in-band channel, a checksum value is sent from a LED-equipped device to a camera or a photosensor-equipped device using light pulses. The size of the checksum varies between 24 bits with an execution time of 5 to 8 seconds and 32 bits with an execution time of 15 seconds. These values are not

consistent with the results in [30] where the authors reimplemented the pairing scheme with a 15-bit OoB message and measured an average completion time equal to 28.8 s

- (2) **KeyLED** [59]: two devices use LED photosensor pair to set up a short-distance visible light communication channel with a raw bit rate of 500 bps and transmit their ECC public keys (352 bits) using on-off keying
- (3) **Flashing displays** [60]: it utilizes two channels, wireless radio as an in-band channel and a unidirectional VLC, where the former is considered as insecure and the latter is used as out-of-band. A VLC is established between the display of a smartphone and a light sensor of a constrained device once it is on top of the screen
- (4) **Secure barcode-based visible light communication (SBVLC)** [61]: a full duplex VLC channel between two camera/display-enabled devices using 2D barcodes. This technique is suitable for device pairing since the main focus of the desired out-of-band channel is the data integrity and not the confidentiality. The barcode can represent the authentication information such as the hashes of the exchanged DH public keys

**2.2.5. Audio.** An audio channel is an acoustic networking system that exploits audible sounds to construct a low-bandwidth communication link using a speaker that generates audio snippets and a microphone that records them, as illustrated in Figure 6. Numerous modulation techniques have been used such as the dual-tone multifrequency (DMTF) and the on-off keying (OOK) to enhance the reliability of the channel.

- (1) **Usability Properties.** The reliability of these channels depends on multiple factors such as the acoustic environment surrounding the devices since the ambient noise drastically increases the transmission errors. Also, the sensitivity of the receiver (microphone) and the distance between the communicating nodes affect the correctness of the signal reception. Based on these factors, the channel requires a human assistance in order to place the devices in a close proximity, to make sure the ambient acoustic environment is suitable for this type of channels and most of all to monitor the

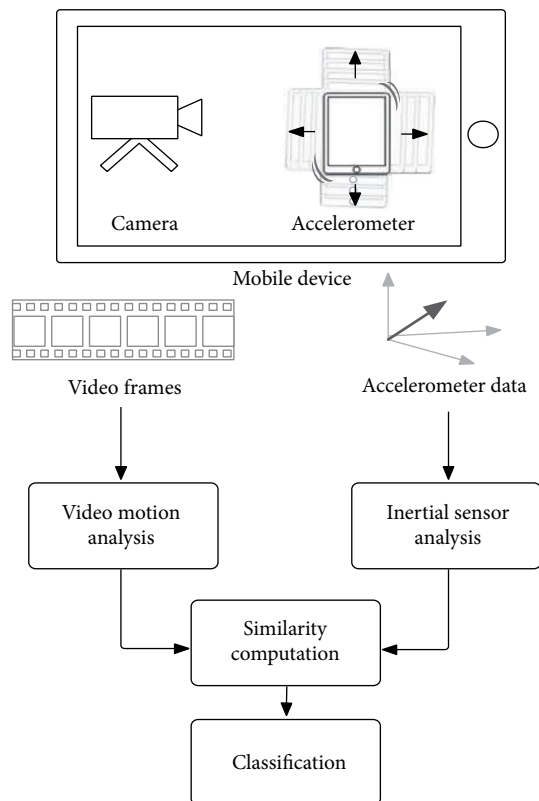


FIGURE 5: Classification of replay attack using video motion analysis and inertial sensor motion analysis [57].

acoustic transfer against any malicious attempt to interfere with the transmission.

(2) *Security Properties*. The feasibility of an eavesdropping makes the confidentiality assumption on these channels out of reach, as demonstrated in the work of Halevi and Saxena [63] using off-the-shelf equipment. Furthermore, the high applicability of a relay attack, as demonstrated in [64], makes the user vigilance during the transmission a necessity.

One of the main advantages of this channel is that an attack is easily detected by a user that is close to the legitimate devices which prevent any active malicious attempts to interfere with the authentication message transmission.

(3) *Proposed Schemes*.

- (1) Loud and clear [65]: the scheme starts by a Diffie-Hellman key exchange over the main insecure channel and then they send the hashes of the public keys encoded in a Mad Lib sentences that are verifiable by the user. Finally, he confirms whether or not the sentences match on both devices. This protocol can also work on a speaker-display-enabled pair of objects where the sentence sent by the speaker of the first one is displayed on the second one
- (2) HAPADEP [35]: the scheme starts by sending the encoded Diffie-Hellman public keys on the audio

channel using fast codec which provides faster transmission rate but it is meaningless to the user. The key verification phase happens also on the audio channel where an audio sequence that is recognizable by the user and that is related to the exchanged public keys is transmitted from each node using slow codec and then they wait for the user to confirm the match

2.2.6. *Haptic*. A haptic channel is constructed using low-frequency mechanical waves that result in a tactile sensation. This type of channel can be either built using only the communicating devices, for example, the use of vibrations to transmit a message [68], as illustrated in Figure 7(a), or it can be a consequence of a user interaction with the objects, for example, by applying a pattern of button presses on the devices [36]. Recently, another variant of SDP protocols has emerged. These schemes rely on the haptic channel that is based on the physical contact between the pairing participants through the body of the user [67, 69], as shown in Figure 7(b). This out-of-band channel is referred to as body-coupled channel (BCC) [70], and this pairing context is also known as wireless body area network (WBAN) or body sensor network (BSN) as detailed in the work of Ali and Khan [23].

(1) *Usability Properties*. The haptic channels tend to demand an extensive user involvement since in most cases he needs to intervene and apply a physical action one or both devices or to monitor any suspicious vibrations coming from an external source.

Also, the use of a vibration motor can be costly when it comes to energy-constrained devices.

However, the fact that the mechanical waves can hardly pass through thick solid objects, such as walls, makes the transmission limited to the physical barriers around the devices, for example, a room. The fact that the communicating objects have to be in direct contact eases the surveillance of the vibrational transfer since the user is only required to focus on the same restricted area.

(2) *Security Properties*. Similar to the audio channels, the confidentiality assumption on these channels no longer holds since they have been proven vulnerable to eavesdropping through acoustic side channel attacks [63]. Due to the necessity of establishing a physical contact between the devices, either by a user intervention or using mechanical waves, the feasibility of an injection attack can be easily detected which guarantees the integrity and the origin authenticity of the exchanged messages. Also, this channel is the only one that is resistant to blocking which makes it the only one that is assuring the liveness property.

(3) *Proposed Schemes*.

- (1) Vibrate-to-unlock [71]: the scheme establishes a secret between a smartphone and an RFID tag using a 14-bit PIN sent through vibration. That secret information, generated by the smartphone, will be required by the tag to identify the legitimate reader

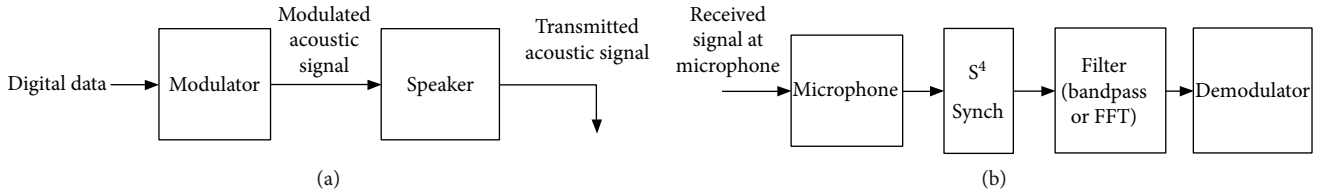


FIGURE 6: Block diagram of an acoustic communication system: (a) modulator/transmitter and (b) demodulator/receiver [62].

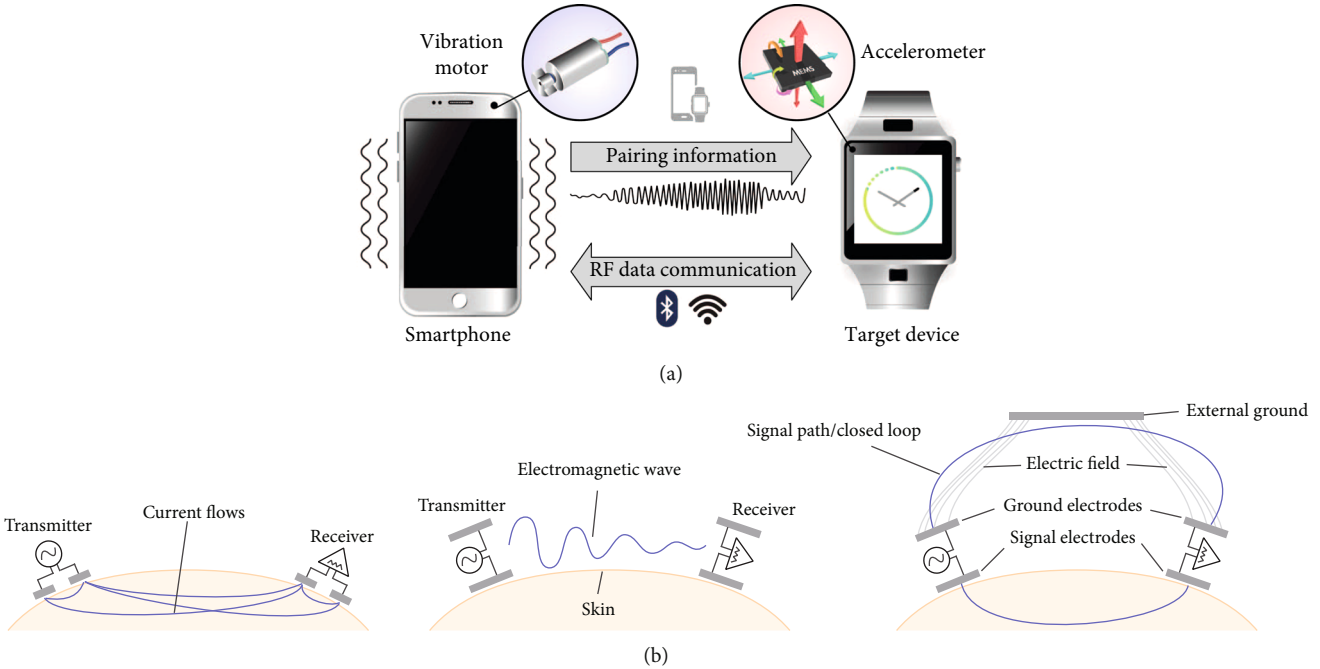


FIGURE 7: Examples of haptic out-of-band channels. (a) Haptic out-of-band channel based on the physical vibrations [66]. (b) Types of body channel communication: galvanic coupling, surface wave, and capacitive coupling [67].

- (2) BEDA [36]: this scheme takes advantage of the user intervention to apply a physical action (button press) on the devices:
- (i) The first variant of this protocol requires the user to establish the same pattern of button presses on both devices (at least seven presses) where these objects will take advantage of the random interevent timing, that is almost equal on each of them, to extract 21 secret bits
  - (ii) The second variant only requires the user to follow a pattern of signals emitted by the first device (pulses of light, vibrations, or beeps) and apply it on the second device using a button. This scheme represents a variant of the protocol MANA III [31] which requires the confidentiality of the PIN entry process. This means that if an adversary is able to witness the pattern of button presses, then he can recompute the 21 secret bits and eventually corrupt the protocol
- (3) Body channel-based secure device pairing [67]: this protocol is based on the capacitive coupling to establish the body communication channel. It has two main phases:
- (i) Key agreement: the two pairing participants establish a secret key  $K$  through the Diffie-Hellman key agreement protocol [72]
  - (ii) Key confirmation: each one of the devices emits a keyed hash of the authentication parameters used through an electrode that is in touch with the human body in order to confirm the correctness of the previous step, as illustrated in Figure 8

### 3. Security Analysis of Out-of-Band Pairing Protocols

**3.1. Threat Model Categories.** In the secure device pairing context, we identify two categories of threat models based on two security properties: *the demonstrative identification* and *the device integrity*. The first property, *the demonstrative identification*, was first introduced in the work of Balfanz et al. [9] and it guarantees the correctness of the pairing initiation process by making sure that the devices performing the pairing are the ones intended to. Therefore, the user plays a crucial part in accomplishing this objective. The second property, *the device integrity*, represents the access privileges

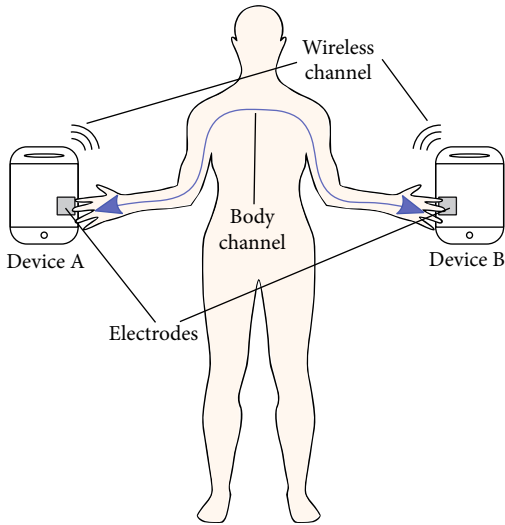


FIGURE 8: Body channel-based secure device pairing [67].

acquired by the attacker on the victim IoT device. Thus, it outlines the fact that one of the pairing participants is partially or completely under the control of the adversary, as detailed in the work of Do et al. [73]. This property covers both the hardware and the firmware integrity of the object. The adopted intruder models, in the formal or the computational security analysis of numerous research works on SDP, assume that the two previously described security properties are achieved. This is explained by the intention to assess the robustness of the scheme by mainly focusing on the protocol exchanges or the employed cryptography. However, the work of Sethi et al. [27] has demonstrated the severity of violating these security requirements by proving the feasibility of an attack that aims at pairing a malicious device instead of a legitimate one. The constraint nature of the target IoT devices is not taken into account in the threat model since we only consider the attacks that are aiming to compromise the pairing procedure. Thus, the denial of sleep [74] or the denial of service attacks [75] are not in the scope of this work. We conclude the existence of two categories of threat models:

**3.1.1. Classical Threat Models.** In this part, the models assume that the demonstrative identification and the device integrity are achieved. This means that the user correctly initiates the pairing between the legitimate participants and that those devices are not under the control of the attacker. To better understand the security analysis of the SDP schemes, outlined in Subsection 3.2, we briefly describe the associated intruder models:

- (i) Dolev-Yao model [6]: in this model, the adversary controls the in-band channel but he has limited capabilities on the out-of-band channel. These limitations are specified by the pairing scheme based on the choice of the channel, as described in Subsection 2.2. However, the cryptographic primitives in this model are considered as a black box and out-of-reach of the adversary. Therefore, the computational

attacks are not assumed feasible. This intruder is also adopted by the strand space model [76]

- (ii) AKISS model [77]: in this model, the capabilities of the adversary are similar to the Dolev-Yao intruder powers. However, the work of Delaune et al. [26] has extended the model to provide the attacker with the capability to guess a low entropy secret
- (iii) Bellare-Rogaway [78, 79]: in this model, each participant is modeled as an oracle that can be addressed by the adversary that allows him to control which party initiates a new pairing session and which participant executes a specific step of the protocol. In addition, the attacker controls the communication between all the participants on the in-band channel and his powers are limited based on the choice of the out-of-band channel, as detailed in Subsection 2.2

**3.1.2. Advanced Threat Model.** In comparison with the initial assumptions of the classical threat model, the demonstrative identification and the device integrity properties in the advanced threat model are not guaranteed.

The former property provides the adversary with the ability to lure the user to initiate the pairing with the wrong device which has been demonstrated feasible and easy to accomplish on the Bluetooth Secure Simple Pairing protocol [27]. Therefore, the correctness of the discovery process of the pairing between the intended devices is affected by the human factor error (HFE) and by the lack of authentication due to the absence of preshared security knowledge.

As for the latter violated property, the adversary is able to gain access to the input/output interfaces of one of the pairing participants which makes him able to intercept any message received by that device without the need of eavesdropping on the in-band or the out-of-band channel. Furthermore, he is able to send any message through that compromised devices which simply makes it an external input/output interface for the attacker. This ability can be achieved either by compromising the hardware [80–83] or the software of the object [84–86].

## 3.2. Security Analysis under the Classical Threat Model

**3.2.1. Description Framework.** The out-of-band-based device pairing protocols have two main building blocks. The first one is the out-of-band channel which constitutes the most important security aspect. The second one is the protocol design that is represented by the cryptographic computations and the exchanges on the in-band channel. In the literature, there are two different aspects when it comes to describing these types of pairing schemes. The first one focuses on the nature of the out-of-band channel by highlighting its communications, security, and usability properties. The second aspect focuses on the protocol design by taking advantage of different cryptographic techniques while abstracting the OoB part to a channel that provides precise security goals as described in Subsection 2.1.

In this part, we will present a selection of OoB-based device pairing protocols that provide a *formal* or a

*computational* security analysis based on the adopted threat model that is described in Subsection 3.1.1. Based on the existing specifications of the chosen research works, we will describe the OoB component using the four main criteria: its nature as stated in Subsection 2.2, its security classification as detailed in our adversary model in Subsection 2.1 and the type of the required user intervention (*relay*, *compare*, *generate* or *set up*) that was first introduced in [3]. Furthermore, we will state the purpose behind the OoB data transmission (*exchange* a parameter, *verify* a value, or *validate* a specific event) since the security requirements on the out-of-band channel are entirely dependent on this information. For example, the use of a confidential channel is only required when the purpose is to exchange a security parameter such as a nonce which is the case for MANA III [31] and MVSec protocols [32].

Finally, we will provide a description framework that represents a summary of the existing security analysis conducted on SDP schemes. This framework will highlight the model used in the analysis: *symbolic* where we assume that the cryptographic primitives used are perfect and we focus entirely on the exchanges or *computational* where we evaluate the cryptographic aspects of the protocol. Also, we will describe the properties evaluated and the outcomes of the verification based on the tested scenarios in the original work. Furthermore, we will assess the results of the analyzed security properties in order to highlight the discovered protocol vulnerabilities that will be, ultimately, used to propose the adequate mitigation. This description framework represents a complete and a systematic approach to describe the two components of the pairing protocol and a clear way of mapping the advantages and limitations of such schemes. The symbols, used in this description, are highlighted in Table 3.

**3.2.2. Evaluated Security Properties.** In the literature, a number of security properties have been evaluated to investigate the correctness of the proposed pairing schemes. However, there is a tendency to provide a different formulation under a different title of the authentication properties that drift away from the commonly known specifications. In order to properly lay out these results and to present a clear overview of these security assessments, we will match the outlined property with the adequate specification in the work of Lowe [24]. However, we will keep the same property formulation as detailed in the original work to provide the reader with a better understanding of the originally conducted security assessment. Based on the definitions in [24], brief descriptions of the assessed security properties are presented as follows:

- (i) Weak agreement: a protocol guarantees to a pairing participant, referred to as Alice, a weak agreement with another participant, referred to as Bob, if, whenever Alice completes a run of the protocol, apparently with Bob, then Bob has previously been executing the protocol, apparently with Alice
- (ii) Injective weak agreement: a protocol guarantees to a pairing participant, referred to as Alice, an injective weak agreement with another participant, referred

TABLE 3: Notations.

Notation	Definition
$\text{mod}$	Modulus operation
$ID_X$	Identifier of the device $X$ (e.g., MAC address)
$\oplus$	Exclusive or operation
$sh(\cdot)$	Short hash function
$sh_K(\cdot)$	Keyed short hash function
$h(\cdot)$	Long hash function
$h_K(\cdot)$	Keyed long hash function using the key $K$
$ X $	Number of bits of $X$
$\hat{x}$	Received value that can be modified by the adversary
$x  y$	Concatenation of the two values $x$ and $y$
$x'$	A value induced by the adversary
$x$	Multiplication operator
$(x \times y)$ – matrix	Matrix with $x$ rows and $y$ columns
$\longrightarrow$	In-band channel
$\dashrightarrow$	Exchange out-of-band channel
$\dashrightarrow$	Verification out-of-band channel
$\dashrightarrow$	Validation out-of-band channel
$Q_X$	The maximum number of sessions launched by the participant $X$
$Q$	The maximum number of sessions launched by any participant

to as Bob, if it guarantees the weak agreement property and, additionally, each protocol run of Alice corresponds to a unique protocol run of Bob

- (iii) Non-injective agreement: the initiator Alice completes a run of the protocol, apparently with Bob, then Bob has previously executed the protocol as a responder, apparently with Alice, and the two parties agreed at the end of the protocol execution on the same parameters
- (iv) Injective agreement: a protocol guarantees to a pairing participant, referred to as Alice, an injective agreement with another participant, referred to as Bob, if it guarantees the noninjective agreement property and, additionally, each protocol run of Alice corresponds to a unique protocol run of Bob

### 3.2.3. Manual Authentication II (MANA II)

(1) *Protocol Steps.* This protocol, proposed by Gehrman et al. [31], is described in Figure 9 and it works as follows:

- (i) ① ② The two devices, named Alice and Bob, exchange their Diffie-Hellman public keys  $g^a$  and  $g^b$  on the in-band

- (ii) ③ ④ The user initiates the authentication process on the device Alice after receiving a confirmation of the public key exchange. This action can be represented as a push button after receiving LED signals from the two objects
- (iii) ⑤ Alice computes a short secret  $K$  (16–20 bits) that is used to generate a short authentication string  $sh_K(g^a || \widehat{g^b})$ .  $sh_K(\cdot)$  represents a one-way function that takes as an argument a short key  $K$  and the concatenation of the DH public keys. Afterwards, she sends it to Bob on the in-band channel
- (iv) ⑥ Alice and Bob display to the user their authentication values,  $K, sh_K(g^a || \widehat{g^b})$  and  $K, sh_K(\widehat{g^a} || g^a)$ , using an output interface (e.g., screen)
- (v) ⑦ The user compares the strings displayed and confirms or rejects the pairing on both devices (e.g., by pressing a button in the case of a successful pairing attempt)

(2) *Out-of-Band Specifications.* The MANA II protocol uses essentially a haptic OoB channel that relies on the physic intervention of the user to compare the displayed messages ③ and ⑥. The purpose of these interactions is to *verify* the short authentication string that is constructed using both the key  $K$  and the short hash function  $sh_K(g^a || g^b)$ . In addition, the same channel is used to *validate* the pairing in message ⑦. The authors assume the use of an authentic channel that guarantees the data freshness, the integrity and the data origin authentication. However, the protocol structure only allows the use of a *delayable-authentic channel* since the adversary is able perform a delay attack on the previous in-band exchanges, as explained in Subsection 2.1, which violate the channel availability property.

(3) *Security Analysis.* The protocol has been formally verified in [25, 26]. The results of the validation are shown in Table 4 and the evaluated security properties are described as follows:

- (1) Paper: Delaune et al. [26]
  - (i) Property description:
    - (a) Non-injective agreement: whenever one of the devices finishes the protocol with the data  $d$  then the other device must have started the protocol with the same data
  - (ii) Assessment: In the original work, the short hash is assumed to be breakable using collision attacks. However, the chosen properties hold over a single session and over two sessions. This is due to the fact that the short authentication key,  $K$ , and the hash of the public DH keys,  $sh_k(g^a || g^b)$ , are both shown to

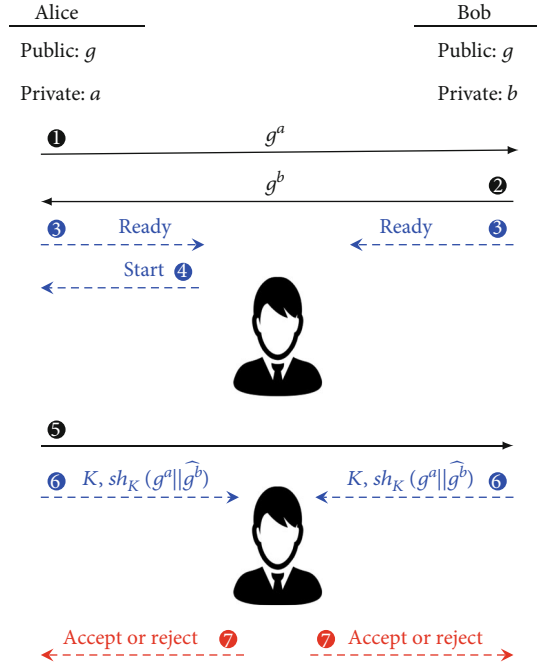


FIGURE 9: Alice and Bob diagram: MANA II protocol.

the user for comparison. This prevents any modification attack that targets any parameters used in the authentication. Therefore, the correctness of the user verification is the only weak link in the authentication process

(2) Paper: Chang and Shmatikov [25]

(i) Properties description:

- (a) Weak agreement: if a device, Alice, successfully completes a protocol execution, apparently with another device Bob, then Bob has executed the protocol at least once and the two participants agreed on their identities
- (b) Injective weak agreement: if a device, Alice, successfully completes a protocol execution, apparently with another device Bob, then Alice has executed the protocol at least once and the two participants agreed on their identities. Additionally, each protocol run of Alice corresponds to a unique protocol run of Bob
- (c) Non-injective agreement: if a device, Alice, successfully completes a protocol execution, apparently with another device Bob, then Alice has executed the protocol at least once and the two participants agreed on all the parameters used to compute the challenge-response values

TABLE 4: Summary of the security proofs.

Protocol	Security analysis	Security analysis model	Security analysis tool	Properties	Tested scenario	Results
MANA II [31]	Delaune et al. [26]	Symbolic	AKISS [77]	Noninjective agreement	Alice to Bob (single session)	Verified
					Bob to Alice (single session)	Verified
					Alice to Bob (two sessions)	Verified
					Bob to Alice (two sessions)	Verified
	Chang and Shmatikov [25]	Symbolic (Dolev-Yao [6])	ProVerif [87]	Injective weak agreement	Alice to Bob	Verified
					Bob to Alice	Verified
					Alice to Bob	Failed
					Bob to Alice	Failed
					Alice to Bob	Failed
					Bob to Alice	Failed
MANA III [31]	Chang and Shmatikov [25]	Symbolic (Dolev-Yao [6])	ProVerif [87]	Key confidentiality	Low entropy PIN	Failed
					Random PIN	Failed
					Low entropy PIN	Verified
					Random PIN	Verified
MANA IV [88] and MA-DH [88]	Laur and Nyberg [88]	Computational	Manual	Upper bound of the successful attack probability	Statistically binding commitment scheme	$2^{-l} + 2\epsilon_1 + 2\epsilon_2 + \epsilon_3$
					Computationally binding commitment scheme	$2^{-l} + 2\epsilon_1 + \epsilon_2 + \sqrt{\epsilon_2} + \epsilon_3$
SAS-based cross-authentication [89]	Vaudenay [89]	Computational (Bellare-Rogaway [78, 79])	Manual	Upper bound of the successful attack probability	One-shot attack	$2^{-l} + \epsilon$
					Multisession attack	$Q_A \times Q_B \times (2^{-l} + \epsilon)$
Improved SAS-based cross-authentication [90]	Pasini and Vaudenay [90]	Computational (Bellare-Rogaway [78, 79])	Manual	Upper bound of the successful attack probability	Multisession attack	$\frac{Q(Q-1)}{2} (2^{-l} + \epsilon + \epsilon_u)$
Ephemeral pairing [91]	Hoepman [91]	Computational (Bellare-Rogaway [78, 79])	Manual	Upper bound of the successful attack probability	Multisession attack	$1 - e^{-Q/2^l} + 2^{- g^d }$
Wong-Stajano asymmetric pairing protocol [92]	Nguyen and Leneutre [93]	Symbolic (strand space model [76])	Manual	Noninjective agreement	Alice to Bob	Failed
					Bob to Alice	Failed
2-round authenticated key agreement protocol [94]	Nguyen and Leneutre [94]	Symbolic (strand space model [76])	Manual	Noninjective agreement	Alice to Bob	Verified
					Bob to Alice	Verified

(d) Injective agreement: if a device, Alice, successfully completes a protocol execution, apparently with another device Bob, then

Alice has executed the protocol at least once and the two participants agreed on all the parameters used to compute the challenge-



response values. Additionally, each protocol run of Alice corresponds to a unique protocol run of Bob

- (ii) **Assessment:** only the weak agreement property holds. This is due to the feasibility of launching multiple protocol executions without binding the session number to the authentication values showed to the user for comparison. This vulnerability leads the human verifier to approve on a pairing process that happened in a second session (tampered with by an attacker) based on the short authentication strings computed over the first session (without any attacker involvement). The protocol should associate a session identifier with the hash displayed to the user in order to mitigate the violations of the authentication properties. The contradiction between the results of the non-injective agreement property is explained by the feasibility of conducting a security verification over an unbounded number by ProVerif [87] of session which is not the case for the AKISS tool [77]

### 3.2.4. Manual Authentication III (MANA III)

(1) *Protocol Steps.* This protocol, proposed by Gehrman et al. [31], is described in Figure 10 and it works as follows:

- (i) ①② The two devices, named Alice and Bob, exchange their Diffie-Hellman public keys  $g^a$  and  $g^b$  on the in-band
- (ii) ③ The user enters a four- to six-digit random number on both devices their input interfaces (e.g., a keypad)
- (iii) ④ Alice computes a long secret  $K_A$  that is used to generate an authentication string  $h_{K_A}(g^a || \widehat{g^b}, R)$ .  $h_K(\bullet)$  which represents a keyed one-way hash function that takes as an argument a long key  $K$ , the concatenation of the DH public keys, and a short nonce  $R$ . Afterwards, she sends it to Bob on the in-band channel
- (iv) ⑤ Bob computes a long secret  $K_B$  that is used to generate an authentication string  $h_{K_B}(\widehat{g^a} || g^b, R)$ .  $h_K(\bullet)$  which represents a keyed one-way hash function that takes as an argument a long key  $K$ , the concatenation of the DH public keys and a short nonce  $R$ . Afterwards, he sends it to Alice on the in-band channel
- (v) ⑥⑦ Alice and Bob exchange the long keys,  $K_A$  and  $K_B$ , on the in-band channel
- (vi) ⑧ Each device notifies the user of the verification outcome (e.g., using an LED signal)

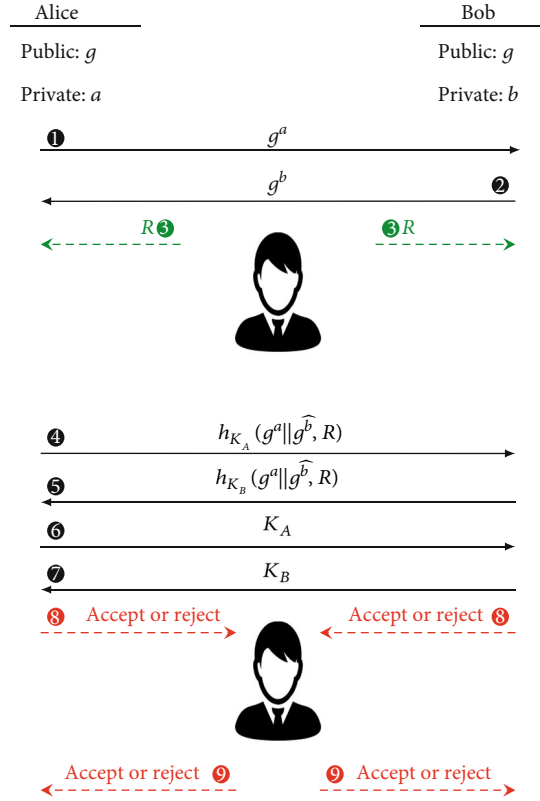


FIGURE 10: Alice and Bob diagram: MANA III protocol.

- (vii) ⑨ The user confirms or rejects the pairing on both devices (e.g., by pressing a button in the case of a successful pairing attempt)

(2) *Out-of-Band Specifications.* The MANA III protocol uses two out-of-band channels that rely on the physical intervention of the user. The first one requires him to *generate* a random PIN  $R$  and to *enter* it in the two pairing devices. This channel is supposed to be out of the reach of the adversary which means that it should be classified as *confidential*. However, the second one only requires the data freshness, the integrity and the data origin authentication. Therefore, this channel is assumed to be classified as *authentic*. On the other hand, the protocol structure only allows the use of *delayable* channels since the adversary is able to perform a delay attack on the previous in-band exchanges, as explained in Subsection 2.1, which violate the channel availability property for both OoB communication links.

(3) *Security Analysis.* The protocol has been formally verified as follows:

- (i) Paper: Chang and Shmatikov [25]
- (ii) Properties description:
  - (1) Key confidentiality: at the end of a successful protocol execution between the two devices, the key is only known to Alice and Bob

(2) Non-injective agreement: if a device, Alice, successfully completes a protocol execution, apparently with another device Bob, then Alice has executed the protocol at least once and the two participants agreed on all the parameters used to compute the challenge-response values

(iii) Assessment: the PIN's confidentiality is a key aspect to accomplish the authentication goal. However, the fact that we rely on the user to provide a random PIN represents a potential vulnerability in the protocol design. This is due to the human tendency to generate a memorable PIN which is easy to guess by the attacker. Therefore, the formal verification of the key secrecy and the non-injective agreement properties does not hold when the PIN has a low entropy. The only solution to guarantee the correctness of the protocol is to use a random PIN that is hard to guess by the attacker. This solution is validated by the formal verification when using a high entropy PIN where both the confidentiality and the authentication goals are achieved

### 3.2.5. Manual Authentication IV (MANA IV) and Manual Authentication Diffie-Hellman (MA-DH)

(1) *Protocol Steps.* This protocol MANA IV, proposed by Laur and Nyberg [88], is described in Figure 11 and it works as follows:

- (i) The two devices, Alice and Bob, generate, respectively, an  $l$ -bit keys,  $k_A$  and  $k_B$ , and their DH private keys,  $a$  and  $b$
- (ii) ① Alice uses a commitment scheme to commit on the key  $k_A$  and sends the commitment and her DH public key  $g^a$  to Bob on the in-band channel
- (iii) ② Bob sends both his DH public key  $g^b$  and the authentication key  $k_B$  to Alice
- (iv) ③ Alice sends her open value  $d_A$  to Bob on the in-band channel
- (v) ④ Alice computes her short authentication string (SAS)  $SAS_A = h_{k_A || k_B}(\widehat{g^a || g^b})$  and sends it to Bob on the out-of-band channel
- (vi) ⑤ Bob verifies the correctness of the SAS sent by Alice and notifies the user to confirm the pairing

In the case of the MA-DH protocol, the authors are using the exchanged Diffie-Hellman public keys for the construction of the authentication string instead of generating the keys,  $k_A$  and  $k_B$ , to avoid the additional computations. The MA-DH protocol structure is described in Figure 12 and it works as follows:

- (i) The two devices, Alice and Bob, generate, respectively, a unique session identifiers,  $ID_A$  and  $ID_B$ ,

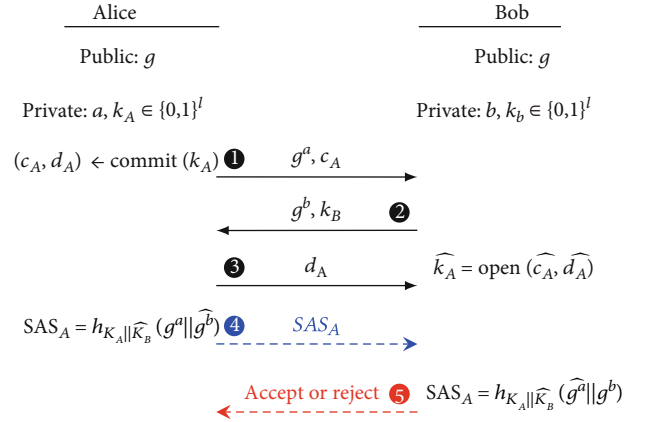


FIGURE 11: Alice and Bob diagram: MANA IV protocol.

and their DH private keys,  $a$  and  $b$ , on the in-band channel

- (ii) Alice uses a commitment scheme to commit on her DH public key  $g^a$  and sends the commitment and her identifier to Bob on the in-band channel
- (iii) Bob sends both his DH public key  $g^b$  and his identifier to Alice on the in-band channel
- (iv) Alice sends her open value  $d_A$  to Bob on the in-band channel
- (v) Alice computes her short authentication string (SAS)  $SAS_A = h_{g^a || g^b}(\widehat{ID_A || ID_B})$  and sends it to Bob on the out-of-band channel
- (vi) Bob verifies the correctness of the SAS sent by Alice and notifies the user to confirm the pairing

(2) *Out-of-Band Specifications.* The MANA IV and the MA-DH protocols are based on the use of two out-of-band channels that have two main purposes: the *verification* of the authentication string and the *validation* of the pairing process. The former channel is required to guarantee the integrity and the data origin authentication without the need for the data freshness property. The security provided is questioned by our adversary model due to the tolerance policy toward replay attacks as detailed in Subsection 2.1. However, the latter channel is required to be classified as *authentic* which makes it hard for the adversary to transmit any messages on the out-of-band. Therefore, we can guarantee the correctness of the validation process. Finally, the structure of protocol allows the attacker to perform a delay attack based on the previous in-band exchanges which violate the channel availability property.

(3) *Security Analysis.* The two protocols have been computationally verified as follows:

- (i) Paper: Laur and Nyberg [88]
- (ii) Verification terminology: Appendix A

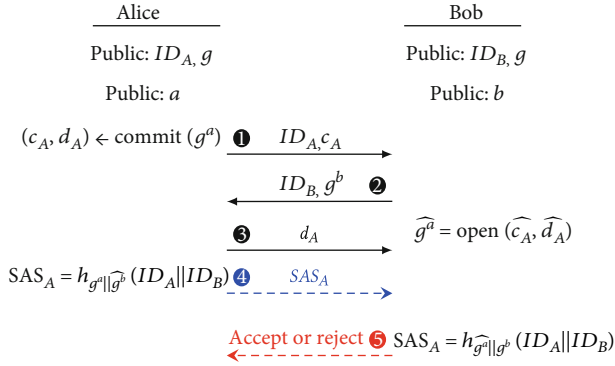


FIGURE 12: Alice and Bob diagram: MA-DH protocol.

(iii) Evaluated properties:

(1) Property: *upper bound of the successful attack probability*(i) Property description: an adversary succeeds in deception if at the end of the protocol Alice and Bob reach the accepting state but  $(g^a, g^{b^\wedge}) \neq (g^{a^\wedge}, g^a)$ . As stated in [88], let  $A$  be the attacker algorithm. A protocol is considered  $(t, \epsilon)$ -secure if for any  $t$ -time attacker  $A$ , the attack success probability is formulated as follows:

$$Adv^{\text{attack}}(A) = \max_{g^a, g^b} \Pr \left[ \text{successful pairing}(g^a, g^{b^\wedge}) \neq (g^{a^\wedge}, g^a) \right] \leq \epsilon. \quad (1)$$

(ii) Tested scenarios:

(a) Statistically binding commitment scheme: for any  $t$ , there exists  $\tau = t + O(1)$  such that if the commit function  $\text{Commit}(\cdot)$  is  $(\tau, \epsilon_1)$ -hiding,  $\epsilon_2$ -binding and  $(\tau, \epsilon_3)$ -nonmalleable and the hash function  $h(\cdot)$  is  $(\epsilon_a, \epsilon_b)$ -almost regular and  $\epsilon_u$ -almost universal then the protocol is  $(2\epsilon_1 + \epsilon_2 + \sqrt{\epsilon_2} + \epsilon_3 + \max\{\epsilon_a, \epsilon_b, \epsilon_u\})$ -secure(b) Computationally binding commitment scheme: for any  $t$ , there exists  $\tau = 2t + O(1)$  such that if the commit function  $\text{Commit}(\cdot)$  is  $(\tau, \epsilon_1)$ -hiding,  $\epsilon_2$ -binding and  $(\tau, \epsilon_3)$ -nonmalleable and the hash function  $h(\cdot)$  is  $(\epsilon_a, \epsilon_b)$ -almost regular and  $\epsilon_u$ -almost universal, then the protocol is  $(2\epsilon_1 + \epsilon_2 + \sqrt{\epsilon_2} + \epsilon_3 + \max\{\epsilon_a, \epsilon_b, \epsilon_u\})$ -secure

(iv) Assessment: the use of a statistically binding commitment scheme provides better security guarantees than the computational one as demonstrated by the

upper bounds of the attack probabilities. Also, it is possible to choose a hash function that provides  $\max\{\epsilon_a, \epsilon_b, \epsilon_u\} = 2^{-l}$ , where  $l$  represents the number of bits sent over the out-of-band channel. Furthermore, it is possible to have a negligible  $\epsilon_1, \epsilon_2$  and  $\epsilon_3$  with respect to the security parameter for a suitable choice of commitment scheme. Thus, MANA IV is considered, based on the definition provided by the original work, asymptotically optimal in terms of security

### 3.2.6. SAS-Based Cross-Authentication Protocol

(4) *Protocol Steps*. This protocol, proposed by Vaudenay [89], is described in Figure 13 and it works as follows:

- (i) The two devices, Alice and Bob, generate, respectively, nonces,  $R_A$  and  $R_B$ , and their DH private keys,  $a$  and  $b$
- (ii) ① Alice uses a commitment scheme to commit on her DH public key  $g^a$  and her nonce  $R_A$ . Then, she sends the commit value  $c_A$  and her public key to Bob on the in-band channel
- (iii) ② Bob uses a commitment scheme to commit on her DH public key  $g^b$  and her nonce  $R_B$ . Then, he sends the commit value  $c_B$  and his public key to Alice on the in-band channel
- (iv) ③ Alice sends her open value  $d_A$  to Bob on the in-band channel
- (v) ④ Bob sends his open value  $d_B$  to Alice on the in-band channel
- (vi) ⑤ Alice retrieves the values hidden in the commitment  $\widehat{c}_B$  sent by Bob using the open value  $\widehat{d}_B$ . He verifies  $b$  both the public key committed and the fact that the first bit is equal to one to avoid reflection attacks. Then, she computes her short authentication string (SAS)  $SAS_A = R_A \oplus \widehat{R}_B$  and sends it to Bob on the out-of-band channel
- (vii) ⑥ Bob verifies the correctness of the SAS sent by Alice and replies with his SAS as a confirmation of the pairing

(5) *Out-of-Band Specifications*. Similar to the MANA IV and MA-DH protocols 3.2.5, the SAS-based cross-authentication scheme is based on the use of two out-of-band channels that have two main purposes: the *verification* of the authentication string and the *validation* of the pairing process. The two channels are required to guarantee the integrity and the data origin authentication without the need for the data freshness property. Therefore, the security provided is questioned by our refined adversary model due to the tolerance policy toward replay attacks as detailed in Subsection 2.1 which can compromise the security of the scheme in a practical scenario. Finally, the structure of protocol allows the attacker to perform a delay attack based on the previous in-band exchanges which violate the channel availability property.

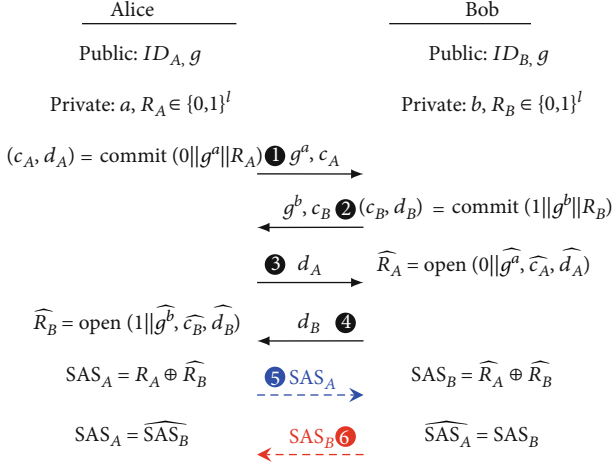


FIGURE 13: Alice and Bob diagram: SAS-based Cross-Authentication protocol.

(6) *Security Analysis.* The protocol has been computationally verified as follows:

- (i) Paper: Vaudenay [89]
- (ii) Verification terminology: Appendix A
- (iii) Evaluated properties:
  - (1) Property: *upper bound of the successful attack probability*
  - (i) Property description: an attack is considered successful if there exists an instance of the protocol, between Alice and Bob, which terminates by reaching an accepting state  $(ID_A \widehat{ID}_B, g^a, g^b) \neq (\widehat{ID}_A, ID_B, g^a, g^b)$
  - (ii) Tested scenarios:
    - (a) One-shot attack: assuming that the commitment scheme is either  $(t_c, \epsilon)$ -extractable or  $(t_c, \epsilon)$ -equivocable, there exists a small constant  $\mu$  (overall time complexity of the protocol) such that for any  $t$ -time adversary,  $P_{\text{one-shot}} \leq 2^{-l} + \epsilon$  or  $t \geq tc - \mu$ , where  $\epsilon$  is negligible
    - (b) Multi-session attack: assuming that  $Q_A$  (respectively,  $Q_B$ ) and  $\mu_A$  (respectively,  $\mu_B$ ) are the maximum number of sessions launched by Alice (respectively, Bob) and the time complexity of the overall authentication protocol for each participant. For any  $t_0$ -time adversary, any  $Q_A$  and  $Q_B$ , the multi-session attack success probability  $P_{\text{multisession}}$  can be formulated using the  $t$ -time one-shot adversary scenario to have  $P_{\text{multisession}} \leq P_{\text{one-shot}} \times Q_A \times Q_B$  with a complexity  $t \leq t_0 + \mu_A \times Q_A + \mu_B \times Q_B$
- (iv) Assessment: the first tested scenario provides the upper bound of the one-shot attack success probabil-

ity. This bound is dependent on the number of bits  $l$  transmitted on the authentication channel and the security parameter  $\epsilon$  of the commitment scheme. Based on the second tested scenario, we can see that the upper bound of the success probability of a multi-session attack can be deduced based on the first result as follows  $P_{\text{multisession}} \leq P_{\text{one-shot}} \times Q_A \times Q_B$ . For a negligible  $\epsilon$ , the probability can be  $Q_A \times Q_B \times 2^{-l}$

### 3.2.7. Improved SAS-Based Cross-Authentication Protocol

(1) *Protocol Steps.* This protocol, proposed by Pasini and Vaudenay [90], is described in Figure 14 and it works as follows:

- (i) The two devices, Alice and Bob, generate, respectively, a hashing key  $K_A$  and a nonce  $R_B$ . Then they generate their DH private keys,  $a$  and  $b$
- (ii) ① Alice uses a commitment scheme to commit on her DH public key  $g^a$  and her hashing key  $K_A$ . Then, she sends the commit value  $c_A$  and her public key to Bob on the in-band channel
- (iii) ② Bob sends his public key  $g^b$  and his nonce  $R_B$  to Alice on the in-band channel
- (iv) ③ Alice sends her open value  $d_A$  to Bob on the in-band channel
- (v) ④ Alice computes her short authentication string (SAS)  $SAS_A = R_A \oplus h_{K_A}(g^{b^a})$  and sends it to Bob on the out-of-band channel
- (vi) ⑤ Bob retrieves the hashing key value from Alice's commitment. Then, he verifies the correctness of the received on the out-of-band channel and replies with his SAS as a confirmation of the pairing

(2) *Out-of-Band Specifications.* Similar to the previous version of this protocol, this improvement is based on the use of two out-of-band channels that have two main purposes: the *verification* of the authentication string and the *validation* of the pairing process. The two channels are required to guarantee the integrity and the data origin authentication without the need for the data freshness property. Therefore, the security provided does not stand based on our refined adversary model due to the tolerance policy toward replay attacks as detailed in Subsection 2.1 which can compromise the security of the scheme in a practical deployment scenario. This tolerance can be further explained by giving the adversary the power to replay previous exchanges but not the ability to inject their own messages under the assumption that we have no preshared secret to construct a signature-based mechanism.

Finally, the structure of protocol allows the attacker to perform a delay attack based on the previous in-band exchanges which violate the channel availability property.

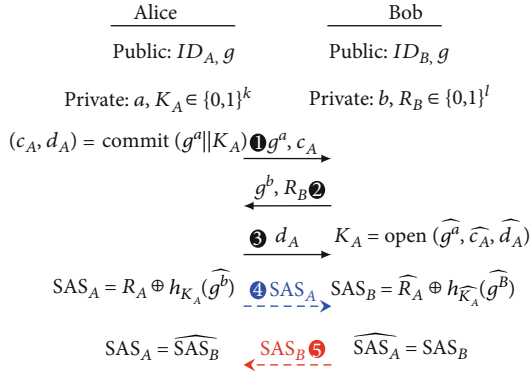


FIGURE 14: Alice and Bob diagram: improved SAS-based cross-authentication protocol.

(3) *Security Analysis*. The protocol has been computationally verified as follows:

- (i) Paper: Pasini and Vaudenay [90]
- (ii) Verification terminology: Appendix A
- (iii) Evaluated properties:
  - (1) Property: *upper bound of the successful attack probability*
- (i) Property description: an attack is considered successful if there exists an instance of the protocol, between Alice and Bob, which terminates by reaching an accepting state  $(ID_A, ID_B, g^a, g^{b^\wedge}) \neq (ID_A, ID_B, g^{a^\wedge}, g^b)$
- (ii) Tested scenario:
  - (a) Multisession attack: let  $\epsilon = q^2 2^{-l_e} + q^2 2^{-l_c}$ , where  $q$  is the maximum number of  $H$  function queries,  $l_e$  is the bit length of the nonce  $e$  used in the random oracle commitment scheme, and  $l_c$  is the bit length of the commit value  $c$ . Let  $h$  be a strongly  $\epsilon_u$ -almost universal hash function with a  $l$ -bit output. The success probability, against an adversary that can launch at maximum  $Q$  instances of Alice or Bob, is bounded by  $(Q(Q-1)/2)(2^{-1}\epsilon + \epsilon_u)$
  - (iv) Assessment: the case of a one-shot success probability attack can be found when assuming  $Q = 2$ . Also, in the work of Laur and Nyberg [88], the extractability and the equivocability notions have been put into question. Furthermore, the use of the Bellare-Rogaway adversary model has been deemed complex and unsuitable for evaluating the security of authentication schemes that run statistically independent consecutive protocol executions (ad hoc device pairing protocols)

3.2.8. *Ephemeral Key Exchange Protocol*. (1) Protocol steps: This protocol, proposed by Hoepman [91], is described in Figure 15 and it works as follows:

- (i) The two devices, Alice and Bob, generate, respectively, their DH private keys,  $a$  and  $b$
  - (ii)  $\textcircled{1}$  Alice commits on her DH public key  $g^a$  using a long hash function  $h(\cdot)$ . Then, she sends the commit value  $h(g^a)$  to Bob on the in-band channel
  - (iii)  $\textcircled{2}$  Bob applies the same computation on his DH public key  $g^b$ . Then, he sends the commit value  $h(g^b)$  to Alice on the in-band channel
  - (iv)  $\textcircled{3}$  Alice sends a short hash of her public key  $sh(g^a)$  to Bob on the out-of-band channel
  - (v)  $\textcircled{4}$  Bob sends a short hash of his public key  $sh(g^b)$  to Alice on the out-of-band channel
  - (vi)  $\textcircled{5}$  Alice sends the real value of her DH public key to Bob on the in-band channel
  - (vii)  $\textcircled{6}$  Bob verifies the two hashes sent in  $\textcircled{1}$  and  $\textcircled{3}$  using the received public key of Alice. Then, he sends the real value of his DH public key on the in-band channel
  - (viii)  $\textcircled{7}$  Alice verifies the two hashes sent in  $\textcircled{2}$  and  $\textcircled{4}$  using the received public key of Bob. Then, she sends a confirmation of the shared DH secret key  $g^{a^\wedge b}$  using the long hash function on the in-band channel
  - (ix)  $\textcircled{8}$  Bob verifies the key confirmation of Alice and confirms the pairing by sending the hash of his DH secret key  $g^{a^\wedge b}$  on the in-band channel
- (2) Out-of-band specifications: the protocol uses a bidirectional out-of-band channel to verify the short hash of the exchanged DH public keys. The channel is supposed to only guarantee the integrity and the origin authentication of the data. Thus, the protocol tolerates any replay attempts by the adversary which might violate the security provided by the scheme when applied to a realistic use-case as detailed in Subsection 2.1. Also, the channel availability property is not guaranteed based on the structure of the protocol
- (3) Security analysis: the protocol has been computationally verified as follows:
- (i) Paper: Hoepman [91]
  - (ii) Verification terminology: Appendix A
  - (iii) Evaluated properties:
    - (1) Property: *upper bound of the successful attack probability*
    - (i) Property description: an attack is considered successful if there exists an instance of the protocol, between Alice and Bob, which

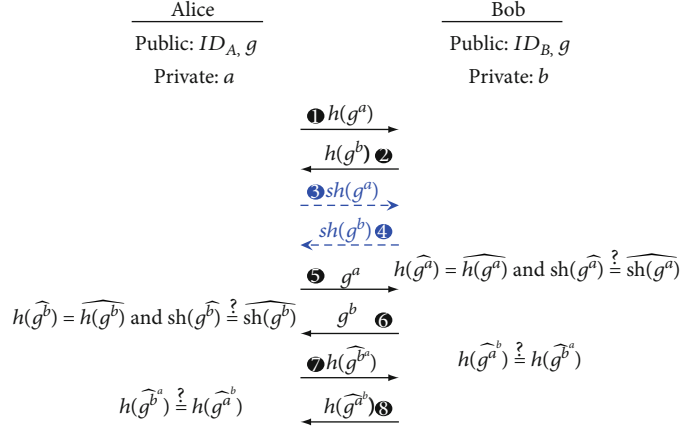


FIGURE 15: Alice and Bob diagram: ephemeral key exchange protocol based on a bidirectional out-of-band channel.

- terminates by reaching an accepting state  $(g^a, g^{b^\wedge}) \neq (g^{a^\wedge}, g^b)$
- (ii) Tested scenario:
- (a) Multi-session attack: let  $l$  be the bit length of the short hash. Let  $Q$  be the maximum number of sessions that can be initiated by the adversary. The successful attack probability is bounded by  $1 - e^{Q/2^l} + 2 - |g^a|$
- (iv) Assessment: the success probability bound has two parts. The first one describes the advantage of an active adversary searching for a collision between the two hashes to bypass the verification. The second part describes the advantage of a passive attacker that tries to guess an  $|g^a|$ -bit DH secret key based on the exchanged public keys. The  $2 \times l$ -bit bidirectional exchanges on the out-of-band channel affect the optimality of the scheme in terms of communication cost since it only provides an attack success probability bound close to  $q \times 2^{-l}$ . This aspect has been improved in the work of Laur and Nyberg [88] where they reduced the number of OoB exchanges by using a single unidirectional channel that only carries a  $t$ -bit authentication string. This improved scheme provides the same level of security by using a single OoB transmission
- (ii) ① Alice sends her identifier  $ID_A$  and her DH public key  $h(g^a)$  to Bob on the in-band channel
- (iii) ② Bob computes the keyed hash  $h_{K_B}(ID_B, R_B, g^b, g^{a^\wedge})$ . Then, he sends it along with his identifier and his DH public key  $g^b$  to Alice on the in-band channel
- (iv) ③ Alice replies by an acknowledgement Ack on the out-of-band channel to confirm the reception of the message ②
- (v) ④ Bob sends the short nonce  $R_B$  to Alice on the out-of-band channel
- (vi) ⑤ Bob sends the value of the hashing key  $K_B$  to Alice on the in-band channel
- (vii) ⑥ Alice verifies the hash sent in using the hashing key and the public key of Bob. Then, she confirms or rejects the pairing on the out-of-band channel
- (2) *Out-of-Band Specifications.* This protocol is based on three out-of-band transmissions that have two main purposes: the *validation* of a specific event and the *exchange* of a parameter related to the authentication process. The two OoB transmissions, ③ and ⑥, require the physical intervention of the user to validate the reception of the message ② by relaying a one-bit interaction to the other device. Thus, these out-of-band channels can be considered haptic, as described in Subsection 2.2.6, which classifies them as *protected* by guaranteeing the integrity, the data origin authenticity, the data freshness, and the liveness properties. As for the out-of-band transmission in message ④, the protocol uses a visible light communication that is classified as *authentic* by providing the integrity, data origin authenticity, and data freshness. Based on the usability analysis conducted in Subsection 2.2.4, the vigilant user is required to set up the devices in a way to create a direct line of sight (LoS). Finally, the protocol structure allows the attacker to delay messages on the out-of-band channel by applying this action on the previous

### 3.2.9. Wong-Stajano Asymmetric Pairing Protocol

(1) *Protocol Steps.* This protocol, proposed by Wong and Stajano [92], is described in Figure 16 and it works as follows:

- (i) The two devices generate, respectively, their DH private keys,  $a$  and  $b$ . Then, Bob generates a short nonce  $R_B$  and long hashing key  $K_B$

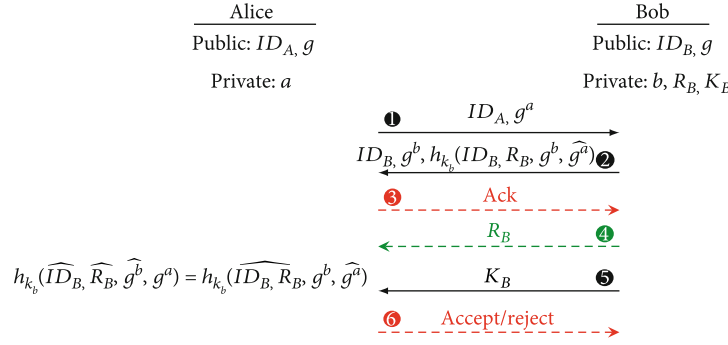


FIGURE 16: Alice and Bob diagram: asymmetric pairing protocol based on a unidirectional out-of-band channel.

in-band exchanges which violate the channel availability property. Therefore, the channels used in this scheme are considered *delayable*.

(3) *Security Analysis*. The protocol has been formally verified as follows:

- (i) Paper: Nguyen and Leneutre [93]
- (ii) Evaluated properties:
  - (a) Property: non-injective agreement [24]
  - (b) Property description: the initiator Alice completes a run of the protocol, apparently with Bob, and then Bob has previously executed the protocol as a responder, apparently with Alice, and the two parties agreed at the end of the protocol execution on the same DH secret key
- (iii) Assessment: the formal analysis has yielded two multi-session attacks that violate the agreement property. These vulnerabilities are based on the delay capability of an attacker over the out-of-band channel and the feasibility of a replay attack that is allowed by the security model of the protocol. This scheme has been improved in the work of Nguyen and Roscoe [5] by eliminating the acknowledgement message which reduces the user intervention. Furthermore, they improved the protocol design by removing the use of two successive unidirectional messages that eliminate the vulnerability noticed by Nguyen and Leneutre [93] later on. From the computational aspect, the new version uses two short nonces and discards the use of a long hashing key which makes it more convenient for the resource-constrained devices

### 3.2.10. 2-Round Authenticated Key Agreement Protocol

(1) *Protocol Steps*. This protocol, proposed by Nguyen and Leneutre [94], is described in Figure 17 and it works as follows:

- (i) The two devices, Alice and Bob, generate, respectively, their DH private keys,  $a$  and  $b$ , and their nonces,  $r_a$  and  $r_b$

- (ii) ① Alice sends her DH public key  $g^a$  and the hash value  $h(g^a, r_a)$  to Bob on the in-band channel
- (iii) ② Bob sends his DH public key  $g^b$  and his nonce  $r_b$  to Alice on the in-band channel
- (iv) ③ Alice computes the value  $r_a \oplus h_{r_b}(g^a, g^b)$  and transfers it to Bob on the out-of-band channel
- (v) ④ Bob retrieves the value of  $r_a$  from the message ③, verifies the hash sent in message ①, and confirms or rejects the pairing on the out-of-band channel

(2) *Out-of-Band Specifications*. This protocol is based on two out-of-band channels that, respectively, serve the purpose of *exchanging* a security parameter related to the authentication process and the purpose of *validating* the pairing. The first channel is supposed to guarantee the integrity and the data origin authenticity without the need for the data freshness property. Thus, the attacker is able to perform a replay on the OoB channel which, according to our security model in Subsection 2.1, might lead to compromising the security of the scheme when deployed in a realistic use-case. The second OoB channel requires the physical intervention of the human operator to relay a one-bit interaction to validate the pairing on the other device. Thus, this haptic channel is classified as *protected* since it guarantees, in addition to the first one, the data freshness and the liveness security properties. Finally, the protocol structure allows the attacker to delay messages on the out-of-band channel by apply this action on the previous in-band exchanges which violate the channel availability property. Therefore, the channels used in this scheme are considered *delayable*

(3) *Security Analysis*. The protocol has been formally verified as follows:

- (i) Paper: Nguyen and Leneutre [94]
- (ii) Evaluated properties:
  - (1) Property: non-injective agreement [24]
  - (a) Property description: the initiator Alice completes a run of the protocol, apparently with

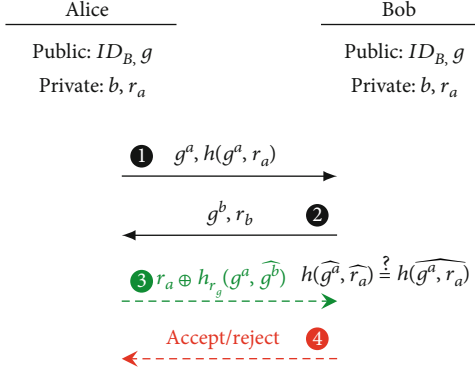


FIGURE 17: Alice and Bob diagram: 2-round authenticated key agreement protocol with unidirectional out-of-band channel.

Bob, and then Bob has previously executed the protocol as a responder, apparently with Alice, and the two parties agreed at the end of the protocol execution on the same DH secret key

(iii)Assessment: based on the manual formal analysis conducted by the authors, the scheme achieves the non-injective agreement property while minimizing the communication costs in terms of number of messages on the in-band and the out-of-band channel. Furthermore, the authors reduced the number of cryptographic primitives to two hash functions without the need to generate another key for hashing in order to comply with the limitations of the resource-constrained devices

**3.2.11. Summary.** In this subsection, we summarize the highlighted results shown in Table 4. The MANA II protocol [31] has been formally verified in [25, 26] using two automated verification tools: ProVerif [87] and AKISS [77]. The work of Delaune et al. [26] focused on evaluating the non-injective agreement property, described in Subsection 3.2.2, under the assumption of having at maximum two protocol sessions. This property holds since the key confirmation step is based on the correctness of a comparison conducted by the user on a short hash displayed by both devices. Thus, any human factor error related to a rush behavior or a one-digit mismatch might compromise the security of the pairing process as detailed in the work of Fomichev et al. [3]. However, a similar formulation of this property has been verified in the work of Chang and Shmatikov [25] based on an unbounded number of sessions. This property does not hold because of the feasibility of launching multiple protocol runs without binding the session number to the short authentication string. Therefore, it is feasible that the user approves a suitable but erroneous authentication value that belongs to previous session. In addition, three other similar formulations of the properties, described in Subsection 3.2.2, have been evaluated: weak agreement, injective weak agreement, and injective agreement. On the first, one holds since it provides the weakest definition authentication by guaranteeing the agreement on the identities of the two intended devices that are assured by their participation in the pairing process. The same work has addressed the confidentiality aspect and the

non-injective agreement of the MANA III protocol [31] based on the assessment of the entropy residing in the PIN that is entered by the user. These results of the verification reflect the importance of having such randomness in the PIN input which is not always the case due to the human tendency to provide a memorable four to six-digit values. On the other hand, the Wong-Stajano asymmetric pairing protocol [92] does not guarantee the non-injective agreement that has been formally evaluated, in the work of Nguyen and Leneutre [93], based on the strand space model [76]. This is due to a vulnerability in the protocol structure against a multi-session attack that exploits the use of two successive unidirectional exchanges which have been corrected in the design proposed in the work of Nguyen and Roscoe [5]. A lightweight pairing scheme has been introduced in another work of Nguyen and Leneutre [94] that achieves formally the previously discussed authentication property using only 4 exchanges. However, this construction is not robust computationally due to the feasibility of a brute-force attack that is aimed at extracting the nonce value from the exchanged hash.

From the computational point of view, the upper bound of the attack success probability of four device pairing schemes has been evaluated. The two variants of the MANA suite protocols, MANA IV and MA-DH [88], have been verified under the assumption of using two different cryptographic primitives: a statistically and a computationally binding commitment schemes. Obviously, the use of the former primitive enhances the security since it reduces the probability bound, but using both constructions, these protocols are asymptotically optimal in terms of security with respect to the number of authentication bits exchanged over the out-of-band channel. The success probability of a multi-session attack on the two short authentication string (SAS) pairing protocols, proposed in [89, 90], has been evaluated under the Bellare-Rogaway model [78, 79]. Nonetheless, in the work of Laur and Nyberg [88], the extractability and the equivocability notions, described in Appendix A, have been questioned along with the use of the Bellare-Rogaway adversary model since it is infeasible to run statistically independent consecutive protocol executions. Finally, the security analysis of the ephemeral pairing scheme, proposed in the work of Hoepman [91], has two outcomes. It describes the advantage of an active adversary searching for a collision between the two hashes to  $a$  bypass the verification. The second part describes the advantage of a passive attacker that tries to guess an  $|g|$ -bit DH secret key based on the exchanged public keys that is usually neglected by the other computational evaluations. On the other hand, the  $2 \times l$ -bit bidirectional exchanges on the out-of-band channel affect the optimality of the scheme in terms of communication cost since it only provides an attack success probability bound close to  $Q \times 2^{-l}$  which has been improved in the work of Laur and Nyberg [88] where they reduced the number of OoB exchanges by using a single unidirectional channel.

### 3.3. Security Analysis under the Advanced Threat Model

**3.3.1. Identity Misbinding Attack.** The identity misbinding attack, also known as *unknown key-share* attack, was first



identified on the station-to-station (STS) protocol [95] in the work of Blake-Wilson and Menezes [96] in 1999. To simplify the attack's applicability on secure device pairing schemes, brought to light in the work of Sethi et al. [27], we will refer to three objects: the legitimate participants Alice and Bob, and the malicious actor Eve. For this attack to work, first, we need to assume that one of the legitimate devices is compromised in a way that lets the attacker control its input and output interfaces. This assumption might be quite strong but it is feasible to introduce a malicious object without being detected especially under the SDP hypothesis of not having any preshared information between the devices. Second, for the attack to work, we need to assume that the identity of the device is determined by the user's physical access to the object such as setting the discovery name on a Bluetooth-enabled device. This assumption is almost always validated since it is the case on the Bluetooth technology that is widely used by the IoT devices.

In Figure 18, we show a misbinding attack during a simple Diffie-Hellman key exchange protocol. Alice initiates the exchange by sending her identifier, represented by her name, and the DH public key  $g^a$ . Eve, our Dolev-Yao intruder, will block the transmission and induce her identifier instead of Alice's. Bob receives the message, identifies the existence of the other device which is Eve, binds her public key to her identifier, computes the secret session key  $K = (g^a)^b = g^{ab}$ , computes the keyed hash  $H_K(g^a, g^b)$ , and finally sends the message Bob,  $g^b, H_K(g^a, g^b)$  to Eve. The attacker replays the same message to Alice that will reply by her own keyed hash to confirm to Bob that she has the same key which was not tampered with. This attack results in a mismatching in the key authentication belief: Alice thinks that she has established a key exchange with Bob, which is technically true, and Bob thinks that he has established a key with a legitimate device that is Eve while hiding completely the existence of Alice. On the other hand, the key confidentiality is not compromised but the key authentication property has been violated.

The presence of an out-of-band channel can solve the issue when the device performing the pairing is not compromised. This is due to the demonstrative identification and data origin authentication properties ensured by the pre-authenticated channel. However, the device's physical integrity is not always granted. Therefore, the risk still needs to be considered for high security level scenarios. Things explain the attack assumption of having at least a compromised device. At this moment, the SDP assumption of having two unidentified devices without any preshared knowledge completely discards the possibility of having any secure binding between the ephemeral session key and the physical objects. Thus, the protocol is vulnerable to any misbinding attempts.

This attack can be more severe when applied against the device pairing schemes. It will not only compromise the key authentication between Alice, the legitimate sound initiator, and Bob, the legitimate compromised device, but also it can lead to pairing Eve with Alice and to neglecting the existence of Bob. This attack is a combination between the unknown key-share, the human error exploitation, and the relay attack.

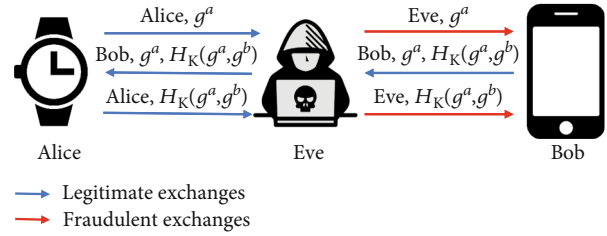


FIGURE 18: Misbinding attack against a Diffie-Hellman key exchange.

In this case, we lure the user to pair Alice with Eve while thinking it is Bob. The attack steps can be detailed as follows:

- (1) Eve uses the same identifier as Bob to maximize the chances of luring the user to initiate the pairing with Eve instead of Bob
- (2) Alice performs a DH key exchange with Eve
- (3) Eve computes the short authentication string (SAS) and sends it to Alice through the out-of-band channel output interface of Bob
- (4) Alice receives the SAS and confirms the pairing to the user

At this stage, the user thinks that Alice and Bob are securely paired while, in fact, he performed the pairing with a malicious object. Therefore, the attacker has succeeded in breaking both the key confidentiality and the key authentication assumptions without the possibility of detecting it.

**3.3.2. Case Study: Bluetooth Secure Simple Pairing (SSP) Protocol.** This attack has been demonstrated on the *numerical comparison* variant of the Bluetooth Secure Simple Pairing (SSP) protocol [42], as shown in Figure 19.

The attack on the SSP protocol can occur as follows:

- (1) The user makes Bob discoverable and starts discovering the neighboring objects enabling Bluetooth
- (2) Eve copies the Bluetooth identifier of Bob and then makes it nondiscoverable
- (3) The user chooses Eve on the list of discoverable devices thinking it was Bob
- (4) Alice and Eve perform the exchanges of the necessary parameters (DH public keys, nonces, commitments...)
- (5) Eve computes the authentication PIN (six-digit verification code) and commands Bob to display it to the user
- (6) Alice computes the authentication PIN and displays it to the user
- (7) The user verifies the match between the two PINs displayed on Alice and Bob
- (8) The user confirms the pairing between Alice and Bob when, in fact, Alice and Eve are paired

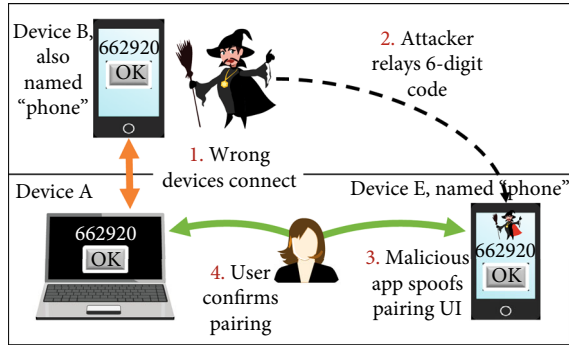


FIGURE 19: Misbinding attack against Bluetooth SSP numeric comparison [97].

The hardest part of the attack, on the SSP protocol, is the feasibility to control the device discovery name by the user. This is due to the necessity of luring the user to initiate the pairing with Eve instead of Bob. This attack can also be conducted on the other two variants of SSP, *PIN entry* and *out-of-band* (using the NFC technology), while excluding the variant *Just Works* since it is not intended for security purposes.

**3.3.3. Case Study: Extensible Authentication Protocol-Nimble Out-of-Band (EAP-NOOB).** This attack can be also applicable to a security bootstrapping protocol under the same assumptions that one participating node is compromised and that the devices identities are defined by the user physical access to them. As an example, the authors of [27] demonstrated this attack on the bootstrapping scheme Extensible Authentication Protocol-Nimble Out-of-Band (EAP-NOOB) [98] that pairs and registers the IoT devices to an online server. This scheme is an authentication method for the Extensible Authentication Protocol [99] that includes an out-of-band channel verification, which requires a degree of user involvement. EAP-NOOB targets the problem of pairing devices without any preshared knowledge and it offers a variety of OoB channels that transfer the authentication string using a QR code, an NFC transmission, or an acoustic exchange. The protocol consists of four main phases:

- (i) In-band key exchange: the IoT object perform an ECDH key exchange with the server
- (ii) Object selection: the user selects the IoT object from a list, provided by the server, on his personal device
- (iii) Out-of-band key authentication: the server sends, on the chosen out-of-band channel, the authentication/i-identification string that authenticates the key exchange and specifically informs the device of its user
- (iv) In-band registration: completes the registration of the device to the user's account on the in-band channel

The misbinding attack, in this case, is aimed at registering a malicious device, called Eve, to the user's account instead of the legitimate but compromised one, referred to as Bob. Fol-

lowing the same example as the one introduced in the original article, Bob will be an object that only has an input interface such as a surveillance camera. The suited out-of-band channel, in this case, is the use of a QR code displayed on the user's personal device (e.g., smartphone).

The attack steps occur as follows:

- (1) The user initiates the pairing by switching on the object Bob
- (2) Bob performs an ECDH key exchange with the server
- (3) The attacker copies Bob's metadata to Eve and initiates the pairing with the server
- (4) The user looks for Bob in a list of the potential devices to be paired that is provided by the server
- (5) The user selects Eve instead of Bob
- (6) The user receives a QR code from the server and shows it to Bob
- (7) Bob sends the QR code to the attacker
- (8) The attacker shows the QR code to Eve
- (9) Eve continues the authentication and the registration process instead of Bob

The hardest part of the attack is luring the user to wrongly select Eve instead of Bob in the second phase of the protocol. Due to this inattentive user behavior, the registration of a malicious device can occur without being noticed using a compromised relay device.

**3.3.4. Mitigation.** The misbinding attack can be mitigated by cryptographically binding the device identifiers to the protocol session. Unfortunately, this solution is not possible for the secure device pairing schemes since the objects do not share any prior information, including preshared symmetric keys or certificates. Another potential solution is the use of copresence verification techniques that are based on variables from the ambient environment. However, numerous samples of these methods have been proven vulnerable against active attacks in the work of Shrestha et al. [64] which does not provide us with a complete solution but it only makes the attack's execution harder on the adversary. Therefore, the mitigation against this attack in the device pairing context is still an open discussion.

## 4. Secure Pairing Design Recommendations and Future Challenges

One of the critical parts of designing a secure device pairing that is based on an out-of-band channel is the assessment of the security guarantees provided by this auxiliary communication medium. This is explained by the absence of any prior knowledge between the pairing parties and the lack of trust in the in-band channel since it is under the control of a powerful Dolev-Yao intruder. Therefore, the OoB channel presents the only source of security in the protocol. As a consequence, if the security properties, assumed guaranteed in

the design phase, are somehow violated by the attacker, then the protocol's security is in jeopardy. The Bluetooth Secure Simple Pairing (SSP) protocol represents one of the most widely used security pairing schemes with its three variants: *PIN entry* inspired from the MANA III protocol, described in Subsection 3.2.4, *numerical comparison* inspired from the MANA II protocol, described in Subsection 3.2.3, and the *out-of-band* which uses the NFC technology 2.2.1. The most deployed ones are *PIN entry* and *numerical comparison*. They rely on the user involvement to either enter a PIN into both devices or to compare and confirm the match between two six-digit number displayed on the objects. Many research works, [100, 101], pointed out numerous vulnerabilities related to the human factor error resulted from the previously described user action, e.g., the entry of a predictable PIN or the confirmation of mismatched authentication digits due to a rush behavior. Another existing design flaw among the secure device pairing schemes is the use of confidential out-of-band channels that are hard to reach due to eavesdropping and side channel attacks. In the work of Han et al. [32], the authors propose a device pairing protocol between a smartphone and a vehicle, called MVSec, that is based on a confidential exchange of a nonce at the beginning of the execution. This confidential channel is unidirectional visible light communication from the car to the device inside the closed glove compartment. According to the attacker model adopted, the adversary can be inside the vehicle and the fact that the light transmission happens inside a close area makes it confidential. Due to the feasibility of the eavesdropping attack using the electromagnetic side channel [102] from a reasonable distance such as an attacker sitting inside the vehicle, the nonce confidentiality assumption no longer holds which compromises the security of the protocol.

The use of the formal or the computational security assessment techniques can be a powerful way to evaluate the confidentiality and the authentication properties provided by the device pairing protocols. However, the only drawback of these methods resides in the formulation of the assessed property that may not reflect the desired degree of security. Therefore, we might end up with an incomplete security analysis or with conflicting results by evaluating two slightly different formulations of the same property as demonstrated in Table 4 in the case of the MANA II protocol. Accordingly, the formulation of these properties should be specified to mitigate the previously discussed issues as detailed in the work of Lowe [24]. Furthermore, the automated formal and computation verification tools should consider the derived categories of the out-of-band channels, highlighted in Section 2, in order to better model their offered security guarantees and to enhance their applicability to the ad hoc secure device pairing protocols. Also, we have noticed that the automated computational analysis using tools such as CryptoVerif [103] does not support the use of out-of-band channels which eliminate the feasibility of performing a complete computational evaluation of numerous device pairing protocols. This is considered as an issue in the device pairing context due to the common use of short authentication strings in the key confirmation phase which is not usually addressed in the symbolic model. Thus, any vulnera-

bilities that exploit the computational weaknesses of the protocol will not be disclosed and, consequently, mitigated. The conducted security evaluations, in both the symbolic and the computational model, demonstrate the necessity of conducting both verifications in order to confirm the resilience of a scheme. This is due to the aspects addressed by each model: the focus on the protocol structure and the exchanges in the symbolic analysis, and also the focus on the computational robustness of the cryptographic primitives. Also, we noticed that the effectiveness of the formal analysis lies in the proper formulation of the security properties under investigation which will, consequently, permit the comparison of the protocol performances. Furthermore, we cannot stress enough the need for a normalized taxonomy in order to enhance the understanding of these security verifications and to better clarify the reasons behind any contradictions between the evaluation outcomes.

Another aspect, that should not be neglect by future work in the secure device pairing field, is the consideration of the advanced threat model, described in Subsection 3.1.2, in the security assessment. Also, there is an imminent need for a possible and a feasible mitigation against this imminent threat using context-based pairing solutions or distance-bounding techniques since the use of out-of-band channels does not provide the necessary security. Finally, with the growing demand for usable and secure device pairing protocol, we noticed the interest in using context-based schemes, also referred to as zero-interaction protocols [2]. However, the security analysis of these techniques is only limited to assessing the randomness of the collected measurements from the ambient environment which reflects the robustness against passive attacks. Such analysis cannot provide the necessary guarantees to formally or computationally validate the security of the pairing procedure as demonstrated in the work of Wu et al. [104] by disclosing a brute-force attack against the interlock protocol applied in the MagPairing protocol [15] that would have been detected using a computational security analysis. Therefore, there is a need for a proper modeling of these pairing schemes based on the security specifications of their chosen contextual features.

## 5. Conclusion

In this survey, we have addressed the secure device pairing problem from the security perspective by providing a refined adversary model on the out-of-band channel that is suitable to the ad hoc pairing context. This threat model eliminates the replay capability of the attacker and it introduces a new notion of delay that is based on the protocol structure rather than the out-of-band channel characteristics. Based on these refinements, we proposed a new out-of-band classification by evaluating a number of security guarantees such as the confidentiality, the data freshness, the integrity, the data authenticity, the liveness, and the channel availability. Furthermore, we surveyed the formal and the computational security analysis conducted on a number of secure device pairing protocols by describing their threat models, their evaluated properties, and their adopted verification models. Although every analysis tends to use its own terminologies

and definitions, we normalized the used taxonomy in order to enhance the understanding of these security verifications and to better clarify the reasons behind any contradictions between the evaluation outcomes. In addition, we discussed the recently published misbinding attack that affects all SDP protocols by exploiting the combination of the lack of hardware protection and the human factor error to lure the user to pair with a malicious device. Our work motivates the use of a formal or a computational security analysis to validate the correctness of the SDP scheme that will be proposed in the future. Our description framework can be extended to all the SDP proposals in the literature in order to create an official secure device pairing repository that clearly describes the security aspects and the discovered attacks on a specific pairing scheme. Finally, we think that the modeling of the out-of-band channels by the security verification tools should be extended in order to better abstract all the security properties guaranteed by these channels that are considered the only source of security in the secure pairing context.

## Appendix

### A. Cryptographic Primitives

In this part, we introduce the properties of the cryptographic primitives used in these security proofs [88–91].

*A.1. Keyed Hash Function.* The keyed hash function  $h : M \times K \rightarrow T$  has two arguments: the first one is the data to be hashed that comes from a word space  $M$  and the second one is the key from a key space  $K$ . This function provides an output in a tag space  $T$  and, depending on the construction of this cryptographic primitive, it can offer the following information theoretic properties:

- (i)  $\epsilon_u$ -almost universal: for any two inputs  $x_0, x_1 \in M$  such that  $x_0 \neq x_1$ , the probability  $\Pr [k \leftarrow K : h(x_0, k) \oplus h(x_1, k)] \leq \epsilon_u$
- (ii)  $\epsilon_u$ -almost XOR universal: for any  $x_0, x_1 \in M$  and  $y \in T$  such that  $x_0 \neq x_1$ , the probability  $\Pr [k \leftarrow K : h(x_0, k) \oplus h(x_1, k) = y] \leq \epsilon_u$ , where  $\epsilon_u \geq 1/|T|$

Also, the notion of almost regular functions has been identified in the case of subkey key manipulation  $h : M \times K_a \times K_b \rightarrow T$ , where  $K_a$  and  $M$  represent the subkey spaces. The following definitions have been introduced:

- (i)  $(\epsilon_a, \epsilon_b)$ -almost regular with respect to the subkeys: for each input  $x \in M, y \in T$  and subkeys  $\widehat{K}_a \in K_a, \widehat{K}_b \in K_b$ , the probabilities  $\Pr [k_a \leftarrow K_a : h(x, k_a, \widehat{K}_b)] \leq \epsilon_a$  and  $\Pr [k_b \leftarrow K_b : h(x, \widehat{K}_a, k_b)] \leq \epsilon_u$ , where  $\epsilon_a, \epsilon_b \geq 1/|T|$
- (ii)  $\epsilon_u$ -almost universal with respect to the subkey  $k_a$ : for any two inputs  $x_0, x_1 \in M$  such that  $x_0 \neq x_1$  and  $k_b, \widehat{k}_b \in K_b$ , the probability  $\Pr [k_a \leftarrow K_a : h(x_0, k_a, k_b) = h(x_1, k_a, \widehat{k}_b)] \leq \epsilon_u$ , where  $\epsilon_u \geq 1/|T|$
- (iii) Strongly  $\epsilon_u$ -almost universal with respect to the subkey  $k_a$ : for any two inputs  $x_0, x_1 \in M$  and  $k_b, \widehat{k}_b \in K_b$  such that  $(x_0, k_b) \neq (x_1, \widehat{k}_b)$ , the probability  $\Pr [k_a \leftarrow K_a : h(x_0, k_a, k_b) = h(x_1, k_a, \widehat{k}_b)] \leq \epsilon_u$ , where  $\epsilon_u \geq 1/|T|$
- (iv) Independence property: let  $x$  be a uniformly distributed variable over the word space  $M$ . Let  $a \in 0, 1^l$  and  $b$  be an arbitrary value from the tag space  $T$ . The two hash functions  $h_1, h_2$  are assumed independent if they satisfy  $\Pr [h_2(x) = a | h_1(x) = b] = \Pr [h_2(x) = a] = 2^{-l}$

*A.2. Commitment Scheme.* The commitment scheme is constructed using three algorithms:

- (i) The generation function  $\text{Gen}$ : generates the public parameters  $pk$  used by the commitment function
- (ii) The commitment function  $\text{Com}_{pk} : M \times R \rightarrow C \times D$ : transforms the input  $m \in M$  and a random value  $r \in R$  into a commitment string  $c \in C$  and an open value  $d \in D$
- (iii) The decommitment function  $\text{Open}_{pk} : C \times D \rightarrow M$ : reveals the value of the commitment string  $m = \text{Open}_{pk}(c, d)$  for all  $(c, d) = \text{Com}_{pk}(m, r)$ . If the algorithm fails to open the commitment, it outputs a special error message  $\perp$

The security of these primitives is defined by a hiding and a binding game. These challenges are conducted against a  $t$  time adversary that tries to violate these properties. The attacker is represented by a function  $A(x_1, \dots, x_n)$  that represents his knowledge  $(x_1, \dots, x_n)$  as inputs to the algorithm. The commitment scheme is  $(t, \epsilon_1)$ -hiding if any  $t$  time adversary achieves the following attack success probability:

$$2 \cdot \left| \Pr [pk \leftarrow \text{Gen}, s \leftarrow \{0, 1\}, \{x_1, x_0\} \leftarrow A(pk), (c_s, d_s) \leftarrow \text{Com}_{pk}(x_s) : A(c_s) = s] - \frac{1}{2} \right| \leq \epsilon_1. \quad (\text{A.1})$$

The commitment scheme is  $(t, \epsilon_2)$ -binding if any  $t$  time adversary achieves the following attack success probability:

$$\Pr \left[ pk \leftarrow \text{Gen}, (c, d_0, d_1) \leftarrow A(pk): \text{Open}_{pk}(c, d_0) \neq \perp \text{ and } \text{Open}_{pk}(c, d_1) \neq \perp \right] \leq \epsilon_2. \quad (\text{A.2})$$

In addition, a commitment scheme is nonmalleable; if given a commitment value  $c$ , the adversary is unable to generate a commitment vector  $(c_1, \dots, c_n)$  that can be opened by a decommitment value  $d$ .

In the work of Pasini and Vaudenay [89, 90], there are two extra commitment properties introduced as follows:

- (i) Extractability: there is a deterministic algorithm  $\text{extract}(m, c)$  that reveals the value of the nonce  $r$  which is hidden along with a message  $m$  in the commitment value  $c = \text{Com}_{pk}(m, r)$  when there exists a decommitment  $d$  such that  $(r, m) = \text{Open}_{pk}(c, d)$
- (ii) Equivocability: there are two deterministic algorithms  $\text{simcommit}(m)$  and  $\text{equivocate}(m, c, r, \phi)$ . The former algorithm returns a fake commitment value  $c$  and an information  $\phi$ . The latter one outputs a decommitment value  $d$  such that we obtain  $(m, r) = \text{Open}_{pk}(c, d)$  from the information  $(c, \phi)$  provided by  $\text{simcommit}$

Furthermore, they use, in [89, 90], the notion of a random oracle commitment scheme where the function  $\text{Com}_{pk}(m, r)$  generates an  $l_e$ -bit value  $e$ , calls a hash function  $H(e, r, m)$ , and outputs the decommitment  $d = (e, r)$ . On the other hand, the decommitment function  $\text{Open}_{pk}(m, c, d)$  simply verifies the hash  $H(d, m) = c$  and uses  $d$  to retrieve  $r$  when the condition holds.

## Data Availability

No data were used to support this study.

## Additional Points

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions can be accessed from [permissions@acm.org](mailto:permissions@acm.org). © 2020 Association for Computing Machinery. Manuscript was submitted to ACM.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the submission of this paper.

## Acknowledgments

This work was supported by the SEIDO LAB (the joint research laboratory for Security and Internet of Things between EDF R&D and Télécom Paris). This research was also funded in part by the National Association for Research and Technology under grant No. 2018/1810.

## References

- [1] M. Conti and C. Lal, "Context-based co-presence detection techniques: a survey," *Computers & Security*, vol. 88, p. 101652, 2019.
- [2] M. Fomichev, M. Maass, L. Almon, A. Molina, and M. Hollick, "Perils of zero-interaction security in the Internet of things," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, pp. 1–38, 2019.
- [3] M. Fomichev, F. Álvarez, D. Steinmetzer, P. Gardner-Stephen, and M. Hollick, "Survey and systematization of secure device pairing," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 517–550, 2017.
- [4] S. Mirzadeh, H. Cruickshank, and R. Tafazolli, "Secure device pairing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 17–40, 2014.
- [5] L. H. Nguyen and A. W. Roscoe, "Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey," *Journal of Computer Security*, vol. 19, no. 1, pp. 139–201, 2011.
- [6] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
- [7] W. R. Claycomb and D. Shin, "Extending formal analysis of mobile device authentication," *Journal of Internet Services and Information Security*, vol. 1, no. 1, pp. 86–102, 2011.
- [8] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.
- [9] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: authentication in ad-hoc wireless networks," in *Proceedings of Network and Distributed System Security Symposium 2002 (NDSS'02)*, San Diego, CA, USA, February 2002.
- [10] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam, "Proximate: proximity-based secure pairing using ambient wireless signals," in *Proceedings of the 9th*

- international conference on Mobile systems, applications, and services - MobiSys '11*, pp. 211–224, Bethesda, Maryland, USA, 2011.
- [11] A. Scannell, A. Varshavsky, A. LaMarca, and E. D. Lara, “Proximity-based authentication of mobile devices,” *International Journal of Security and Networks*, vol. 4, no. 1/2, pp. 4–16, 2009.
- [12] W. Xi, C. Qian, J. Han et al., “Instant and robust authentication and key agreement among mobile devices,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 616–627, Vienna, Austria, October 2016.
- [13] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, “Sound-proof: usable two-factor authentication based on ambient sound,” in *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pp. 483–498, Washington D.C, USA, 2015.
- [14] D. Schürmann and S. Sigg, “Secure communication based on ambient audio,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 358–370, 2013.
- [15] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, “Mag-Pairing: pairing smartphones in close proximity using magnetometers,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1306–1320, 2016.
- [16] M. Miettinen, N. Asokan, T. D. Nguyen, A.-R. Sadeghi, and M. Sobhani, “Context-based zero-interaction pairing and key evolution for advanced personal devices,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*, Scottsdale, Arizona, USA, 2014.
- [17] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan, “Drone to the rescue: relay-resilient authentication using ambient multi-sensing,” in *Financial Cryptography and Data Security*, Springer, 2014.
- [18] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, and P. Nurmi, “Comparing and fusing different sensor modalities for relay attack resistance in zero-interaction authentication,” in *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Budapest, Hungary, March 2014.
- [19] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen, “OPFKA: secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks,” in *2013 Proceedings IEEE INFOCOM*, pp. 2274–2282, Turin, Italy, April 2013.
- [20] Q. Jiang, Z. Chen, J. Ma, X. Ma, J. Shen, and D. Wu, “Optimized fuzzy commitment based key agreement protocol for wireless body area network,” *IEEE Transactions on Emerging Topics in Computing*, p. 1, 2019.
- [21] D. Oberoi, W. Y. Sou, Y. Y. Lui, R. Fisher, L. Dinca, and G. P. Hancke, “Wearable security: key derivation for body area sensor networks based on host movement,” in *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, pp. 1116–1121, Santa Clara, CA, USA, June 2016.
- [22] D. Schürmann, A. Brüschi, S. Sigg, and L. Wolf, “Bandana—body area network device-to-device authentication using natural gait,” in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 190–196, Kona, HI, USA, 2017.
- [23] A. Ali and F. A. Khan, “Key agreement schemes in wireless body area networks: taxonomy and state-of-the-art,” *Journal of Medical Systems*, vol. 39, no. 10, p. 115, 2015.
- [24] G. Lowe, “A hierarchy of authentication specifications,” in *Proceedings 10th Computer Security Foundations Workshop*, pp. 31–43, Rockport, MA, USA, 1997.
- [25] R. Chang and V. Shmatikov, “Formal analysis of authentication in Bluetooth device pairing,” *FCS-ARSPA07*, p. 45, 2007.
- [26] S. Delaune, S. Kremer, and L. Robin, “Formal verification of protocols based on short authenticated strings,” in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 130–143, Santa Barbara, CA, USA, August 2017.
- [27] M. Sethi, A. Peltonen, and T. Aura, “Misbinding attacks on secure device pairing and bootstrapping,” in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pp. 453–464, Auckland, New Zealand, July 2019.
- [28] L. F. Cranor, “A framework for reasoning about the human in the loop,” in *UPSEC'08: Proceedings of the 1st Conference on Usability, Psychology, and Security*, pp. 1:1–1:15, Berkeley, CA, USA, April 2008.
- [29] R. Kainda, I. Flechais, and A. W. Roscoe, “Usability and security of out-of-band channels in secure device pairing protocols,” in *Proceedings of the 5th Symposium on Usable Privacy and Security - SOUPS '09*, p. 11, Mountain View, California, USA, 2009.
- [30] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun, “A comparative study of secure device pairing methods,” *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 734–749, 2009.
- [31] C. Gehrmann, C. J. Mitchell, and K. Nyberg, “Manual authentication for wireless devices,” *RSA Cryptobytes*, vol. 7, no. 1, pp. 29–37, 2004.
- [32] J. Han, Y.-H. Lin, A. Perrig, and F. Bai, “Short paper: MVSec: secure and easy-to-use pairing of mobile devices with vehicles,” in *Proceedings of the 2014 ACM conference on Security and privacy in wireless & mobile networks - WiSec '14*, pp. 51–56, Oxford, United Kingdom, 2014.
- [33] J. Sen, “Security in wireless sensor networks,” *Wireless Sensor Networks: Current Status and Future Trends*, vol. 407, p. 408, 2012.
- [34] B. Alpern and F. B. Schneider, “Recognizing safety and liveness,” *Distributed Computing*, vol. 2, no. 3, pp. 117–126, 1987.
- [35] C. Soriente, G. Tsudik, and E. Uzun, “HAPADEP: human-assisted pure audio device pairing,” in *Information Security*, pp. 385–400, Springer, 2008.
- [36] C. Soriente and E. Uzun, “BEDA: button-enabled device association”.
- [37] “How NFC works,” nearfieldcommunication.org, 2015, <http://nearfieldcommunication.org/how-it-works.html>.
- [38] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, “Practical NFC peer-to-peer relay attack using mobile phones,” in *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, pp. 35–49, Springer, 2010.
- [39] R. Zhou and G. Xing, “nShield: a noninvasive NFC security system for mobile devices,” in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services - MobiSys '14*, pp. 95–108, 2014.
- [40] S. Akter, T. Chakraborty, T. A. Khan, S. Chellappan, and I. Al, “Can you get into the middle of near field communication?,” in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, pp. 365–373, Singapore, Singapore, October 2017.
- [41] W.-F. Alliance, *Wi-Fi Simple Configuration Technical Specification, version 2.0. 5*, Wi-Fi Alliance, 2014.

- [42] S. Bluetooth, *Bluetooth Core Specification v5. 0*, Bluetooth Special Interest Group, Kirkland, WA, USA, 2016.
- [43] S. A. Weis, "RFID (radio frequency identification): principles and applications," *System*, vol. 2, no. 3, pp. 1–23, 2007.
- [44] N. Instrument, *Advanced RFID Measurements: Basic Theory to Protocol Conformance Test*, 2013.
- [45] C. Castelluccia and G. Avoine, "Noisy tags: a pretty good key exchange protocol for RFID tags," in *International Conference on Smart Card Research and Advanced Applications*, pp. 289–299, Springer, 2006.
- [46] K.-C. Huang and Z. Wang, *Millimeter Wave Communication Systems*, vol. 29, John Wiley & Sons, 2011.
- [47] IEEE Standards Association et al., *IEEE Std 802.11 ad-2012, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (phy) Specifications," Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band, IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems—Local and Metropolitan Area Networks—Specific Requirements*, IEEE Computer Society, 2012.
- [48] W.-F. Alliance, *WiGig® and the Future of Seamless Connectivity*, Wi-Fi Alliance, 2013.
- [49] A. Oncu and M. Fujishim, "Millimeter-wave CMOS impulse radio," in *Advances in Solid State Circuit Technologies*, pp. 255–288, 2010.
- [50] D. Steinmetzer, J. Chen, J. Classen, E. Knightly, and M. Hollick, "Eavesdropping with periscopes: experimental security analysis of highly directional millimeter waves," in *2015 IEEE Conference on Communications and Network Security (CNS)*, pp. 335–343, Florence, Italy, September 2015.
- [51] *SiBEAM captures world's first 60 GHz millimeter-wave smartphone design win in LeTV's flagship smartphone*, *Le Max*, 2015.
- [52] *HP Elite x2 1011 G2 - connecting to the wireless dock*, 2019.
- [53] A. R. Ndjiongue, H. C. Ferreira, and T. M. Ngatched, *Visible Light Communications (VLC) Technology*, Wiley Encyclopedia of Electrical and Electronics Engineering, 1999.
- [54] S. D. Perli, N. Ahmed, and D. Katabi, "Pixnet: interference-free wireless links using LCD-camera pairs," in *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking - MobiCom '10*, pp. 137–148, Chicago, Illinois, USA, 2010.
- [55] J. Classen, J. Chen, D. Steinmetzer, M. Hollick, and E. Knightly, "The spy next door: eavesdropping on high throughput visible light communications," in *Proceedings of the 2nd International Workshop on Visible Light Communications Systems*, pp. 9–14, Paris, France, 2015.
- [56] A. Dziech, J. Bialas, A. Glowacz et al., "Overview of recent advances in CCTV processing chain in the INDECT and INSIGMA projects," in *2013 International Conference on Availability, Reliability and Security*, pp. 836–843, Regensburg, Germany, September 2013.
- [57] M. Rahman, U. Topkara, and B. Carburnar, "Seeing is not believing: visual verifications through liveness analysis using mobile devices," in *Proceedings of the 29th Annual Computer Security Applications Conference on - ACSAC '13*, pp. 239–248, New Orleans, Louisiana, USA, 2013.
- [58] N. Saxena, J.-E. Ekberg, K. Kostianen, and N. Asokan, "Secure device pairing based on a visual channel," in *2006 IEEE Symposium on Security and Privacy (S&P'06)*, p. 6, Berkeley/Oakland, CA, USA, 2006.
- [59] R. Roman and J. Lopez, "KeyLED-transmitting sensitive data over out-of-band channels in wireless sensor networks," in *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pp. 796–801, Atlanta, GA, USA, 2008.
- [60] T. Kovačević, T. Perković, and M. Čagalj, "Flashing displays: user-friendly solution for bootstrapping secure associations between multiple constrained wireless devices," *Security and Communication Networks*, vol. 9, no. 10, 2016.
- [61] B. Zhang, K. Ren, G. Xing, X. Fu, and C. Wang, "SBVLC: secure barcode-based visible light communication for smartphones," *IEEE Transactions on Mobile Computing*, vol. 15, no. 2, pp. 432–446, 2016.
- [62] R. Jurdak, A. G. Ruzzelli, G. M. P. O'Hare, and C. V. Lopes, "Mote-based underwater sensor networks: opportunities, challenges, and guidelines," *Telecommunication Systems*, vol. 37, no. 1-3, pp. 37–47, 2008.
- [63] T. Halevi and N. Saxena, "Acoustic eavesdropping attacks on constrained wireless device pairing," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 563–577, 2013.
- [64] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan, "Sensor-based proximity detection in the face of active adversaries," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 444–457, 2019.
- [65] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, "Loud and clear: human-verifiable authentication based on audio," in *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, p. 10, Lisboa, Portugal, 2006.
- [66] K. Lee, V. Raghunathan, A. Raghunathan, and Y. Kim, "Sync-Vibe: fast and secure device pairing through physical vibration on commodity smartphones," in *2018 IEEE 36th International Conference on Computer Design (ICCD)*, pp. 234–241, Orlando, FL, USA, October 2018.
- [67] M. Roeschlin, I. Martinovic, and K. B. Rasmussen, "Device pairing at the touch of an electrode," *NDSS*, vol. 18, pp. 18–21, 2018.
- [68] N. Roy, M. Gowda, and R. R. Choudhury, "Ripple: communicating through physical vibration," in *12th {USENIX} Symposium on Networked Systems Design and Implementation (NSDI'15)*, pp. 265–278, Oakland, CA, USA, 2015.
- [69] M. Li, S. Yu, W. Lou, and K. Ren, "Group device pairing based secure sensor association and key management for body area networks," in *2010 Proceedings IEEE INFOCOM*, pp. 1–9, San Diego, CA, USA, March 2010.
- [70] A. Ruaro, J. Thaysen, and K. B. Jakobsen, "Head-centric body-channel propagation paths characterization," in *2015 9th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–4, Berlin, Heidelberg, Germany, 2015.
- [71] N. Saxena, M. B. Uddin, J. Voris, and N. Asokan, "Vibrate-to-unlock: mobile phone assisted user authentication to multiple personal RFID tags," in *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 181–188, Seattle, WA, USA, March 2011.
- [72] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 2006.
- [73] Q. Do, B. Martini, and K.-K. R. Choo, "The role of the adversary model in applied security research," *Computers & Security*, vol. 81, pp. 156–181, 2019.
- [74] A. Gallais, T.-H. Hedli, V. Loscri, and N. Mitton, "Denial-of-sleep attacks against IoT networks," in *2019 6th International*

- Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1025–1030, Paris, France, April 2019.
- [75] C. Koliass, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and other botnets,” *Computer*, vol. 50, no. 7, pp. 80–84, 2017.
- [76] F. J. T. Fábrega, J. C. Herzog, and J. D. Guttman, “Strand spaces: why is a security protocol correct?,” in *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No.98CB36186)*, pp. 160–171, Oakland, CA, USA, 1998.
- [77] R. Chadha, V. Cheval, Ş. Ciobăcă, and S. Kremer, “Automated verification of equivalence properties of cryptographic protocols,” *ACM Transactions on Computational Logic (TOCL)*, vol. 17, no. 4, pp. 1–32, 2016.
- [78] M. Bellare and P. Rogaway, “Entity authentication and key distribution,” in *Advances in Cryptology — CRYPTO’ 93. CRYPTO 1993. Lecture Notes in Computer Science, vol 773*, D. R. Stinson, Ed., pp. 232–249, Springer, Berlin, Heidelberg, 1994.
- [79] M. Bellare and P. Rogaway, “Provably secure session key distribution: the three party case,” in *STOC ’95: Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pp. 57–66, Las Vegas, NV, USA, May 1995.
- [80] J. Clark, S. Leblanc, and S. Knight, “Compromise through USB-based hardware Trojan horse device,” *Future Generation Computer Systems*, vol. 27, no. 5, pp. 555–563, 2011.
- [81] J. Dofe, J. Frey, and Q. Yu, “Hardware security assurance in emerging IoT applications,” in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2050–2053, Montreal, QC, Canada, May 2016.
- [82] S. Moein, F. Gebali, and I. Traore, “Analysis of covert hardware attacks,” *Journal of Convergence*, vol. 5, 2014.
- [83] N. Patwari, J. Croft, S. Jana, and S. K. Kasera, “High-rate uncorrelated bit extraction for shared secret key generation from channel measurements,” *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 17–30, 2010.
- [84] H. A. Abdul-Ghani, D. Konstantas, and M. Mahyoub, “A comprehensive IoT attacks survey based on a building-blocked reference model,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, 2018.
- [85] M. M. Ahemd, M. A. Shah, and A. Wahid, “IoT security: a layered approach for attacks defenses,” in *2017 International Conference on Communication Technologies (ComTech)*, pp. 104–110, Rawalpindi, Pakistan, April 2017.
- [86] J. Deogirikar and A. Vidhate, *Security Attacks in IoT: A Survey* 32–37.
- [87] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, *ProVerif 2.00: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*, pp. 05–16, 2018.
- [88] S. Laur and K. Nyberg, “Efficient mutual data authentication using manually authenticated strings,” in *Cryptology and Network Security*, pp. 90–107, Springer, 2006.
- [89] S. Vaudenay, “Secure communications over insecure channels based on short authenticated strings,” in *Advances in Cryptology – CRYPTO 2005*, pp. 309–326, Springer, 2005.
- [90] S. Pasini and S. Vaudenay, “SAS-based authenticated key agreement,” in *Public Key Cryptography - PKC 2006*, pp. 395–409, Springer, 2006.
- [91] J.-H. Hoepman, “The ephemeral pairing problem,” in *Financial Cryptography*, pp. 212–226, Springer, 2004.
- [92] F. L. Wong and F. Stajano, “Multichannel security protocols,” *IEEE Pervasive Computing*, vol. 6, no. 4, pp. 31–39, 2007.
- [93] T. Nguyen and J. Leneutre, “Formal analysis of secure device pairing protocols,” in *2014 IEEE 13th International Symposium on Network Computing and Applications*, pp. 291–295, Cambridge, MA, USA, August 2014.
- [94] T. Nguyen and J. Leneutre, “A secure and effective device pairing protocol,” in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 507–512, Las Vegas, NV, USA, 2015.
- [95] W. Diffie, P. C. Van Oorschot, and M. J. Wiener, “Authentication and authenticated key exchanges,” *Designs, Codes and Cryptography*, vol. 2, no. 2, pp. 107–125, 1992.
- [96] S. Blake-Wilson and A. Menezes, “Unknown key-share attacks on the station-to-station (STS) protocol,” in *Public Key Cryptography. PKC 1999. Lecture Notes in Computer Science, vol 1560* pp. 154–170, Springer, Berlin, Heidelberg.
- [97] A. Peltonen, M. Sethi, and T. Aura, “Formal verification of misbinding attacks on secure device pairing and bootstrapping,” *Journal of Information Security and Applications*, vol. 51, article 102461, 2020.
- [98] T. Aura and M. Sethi, *Nimble Out-of-Band Authentication for EAP (EAP-NOOB). draft-aura-eap-noob-03 (Work in Progress)*, 2018.
- [99] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowitz, “et al,” *Extensible authentication protocol (EAP)*, 2004.
- [100] C. Kuo, J. Walker, and A. Perrig, “Low-cost manufacturing, usability, and security: an analysis of Bluetooth simple pairing and Wi-Fi protected setup,” in *Financial Cryptography and Data Security*, pp. 325–340, Springer, 2007.
- [101] D.-Z. Sun, Y. Mu, and W. Susilo, “Man-in-the-middle attacks on secure simple pairing in Bluetooth standard v5. 0 and its countermeasure,” *Personal and Ubiquitous Computing*, vol. 22, no. 1, pp. 55–67, 2018.
- [102] H. Tanaka, “Information leakage via electromagnetic emanations and evaluation of tempest countermeasures,” in *Information Systems Security*, pp. 167–179, Springer, 2007.
- [103] B. Blanchet, “CryptoVerif: computationally sound mechanized prover for cryptographic protocols,” *Dagstuhl Seminar Formal Protocol Verification Applied*, vol. 117, p. 156, 2007.
- [104] Y. Wu, B. Chen, Z. Zhao, and Y. Cheng, “Attack and countermeasure on interlock-based device pairing schemes,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 3, pp. 745–757, 2018.



## Research Article

# Determining the Image Base of Smart Device Firmware for Security Analysis

Ruijin Zhu <sup>1</sup>, Baofeng Zhang,<sup>1,2</sup> Yu-an Tan,<sup>3</sup> Jinmiao Wang <sup>4,5</sup> and Yueliang Wan<sup>4,5</sup>

<sup>1</sup>China Information Technology Security Evaluation Center, Beijing 100085, China

<sup>2</sup>Tsinghua University, Beijing 100084, China

<sup>3</sup>School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

<sup>4</sup>Run Technologies Co., Ltd. Beijing, Beijing 100192, China

<sup>5</sup>Beijing Engineering Research Center for Cyberspace Data Analysis and Applications, Beijing 100083, China

Correspondence should be addressed to Jinmiao Wang; [jinmiao\\_wang@163.com](mailto:jinmiao_wang@163.com)

Received 28 June 2020; Revised 7 September 2020; Accepted 24 September 2020; Published 28 December 2020

Academic Editor: Ding Wang

Copyright © 2020 Ruijin Zhu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The authorization mechanism of smart devices is mainly implemented by firmware, yet many smart devices have security issues about their firmware. Limited research has focused on securing the firmware of smart devices, although increasingly more smart devices are used to deal with the very sensitive applications, activities, and data of users. Thus, research on smart device firmware security is of growing importance. Disassembly is a common method for evaluating the security of authorization mechanisms. When disassembling firmware, the processor type of the running environment and the image base of the firmware should first be determined. In general, the processor type can be obtained by tearing down the device or consulting the product manual. However, it is not easy to determine the image base of firmware. Since the processors of many smart devices are ARM architectures, in this paper, we focus on firmware under the ARM architecture and propose an automated method for determining the image base. By studying the storage law of the jump table in the firmware of ARM-based smart devices, we propose an algorithm, named determining the image base by searching jump tables (DBJT), to determine the image base. The experimental results indicate that the proposed method can successfully determine the image base of firmware, which stores the absolute address in the jump table.

## 1. Introduction

Wireless technologies for smart devices are developing rapidly and are widely used. Smart devices have been deployed in several scenarios, such as smart phones, wearable devices, and vehicles. A recent marketing research report forecasted that the amount of smart devices will grow to approximately 10 billion in number worldwide by 2025 [1].

There have been a number of authorization security incidents caused by defects in firmware in recent years. For example, researchers found that several D-Link routers contain authentication backdoors by disassembling the firmware. If the attacker's browser user agent string is `xmlset_roodkcableoj28840ybtide`, then he/she can access the web interface of the device, bypassing the authentication procedure and viewing/changing the device settings [2]. A similar

incident occurred on the Tenda router, in which an authentication backdoor was found by disassembling the firmware. The backdoor allows for the execution of commands remotely by sending them to specific strings and commands [3].

Unlike traditional embedded devices, smart devices are more vulnerable to attack. Some incidents [4–8] indicate that the security situation of smart devices is becoming increasingly serious, which has a profound impact on a country's economic and social development. Therefore, the security evaluation analysis and vulnerability assessment of smart devices are the primary considerations at present.

However, limited papers have been found that focus on securing the firmware of smart devices, although the firmware running on these smart devices is vulnerable to attack. Firmware provides the necessary instructions on how a smart device determines its functionality and communicates with

```

ROM:00000950 sub_950
ROM:00000950
ROM:00000950 var_18 = -0x18
ROM:00000950
ROM:00000950 STMFD SP!, {R4,R5,LR}
ROM:00000954 SUB SP, SP, #0xC
ROM:00000958 LDR R5, =0xC00310A8
ROM:0000095C LDR R4, [R5]
ROM:00000960 CMP R4, #0
ROM:00000964 BNE loc_99C
ROM:00000968 LDR R1, =0xC00310AC
ROM:0000096C MOV R2, #0x1000
ROM:00000970 LDR R0, =0xC00300A8
ROM:00000974 BL sub_111180
ROM:00000978 LDR R12, =0xC0018E3C
ROM:0000097C MOV R3, R4
ROM:00000980 MOV R2, R4
ROM:00000984 LDR R0, =0xC023CF6E
ROM:00000988 LDR R1, =0xC00300A8
ROM:0000098C STR R12, [SP,#0x18+var_18]
ROM:00000990 BL sub_53354
ROM:00000994 MOV R3, #1
ROM:00000998 STR R3, [R5]

```

(a) The image base is set to 0

```

ROM:C0018950 sub_C0018950
ROM:C0018950
ROM:C0018950 var_18 = -0x18
ROM:C0018950
ROM:C0018950 STMFD SP!, {R4,R5,LR}
ROM:C0018954 SUB SP, SP, #0xC
ROM:C0018958 LDR R5, =dword_C00310A8
ROM:C001895C LDR R4, [R5]
ROM:C0018960 CMP R4, #0
ROM:C0018964 BNE loc_C001899C
ROM:C0018968 LDR R1, =unk_C00310AC
ROM:C001896C MOV R2, #0x1000
ROM:C0018970 LDR R0, =unk_C00300A8
ROM:C0018974 BL sub_C0129180
ROM:C0018978 LDR R12, =sub_C0018E3C
ROM:C001897C MOV R3, R4
ROM:C0018980 MOV R2, R4
ROM:C0018984 LDR R0, =aEarlyOptions
ROM:C0018988 LDR R1, =unk_C00300A8
ROM:C001898C STR R12, [SP,#0x18+var_18]
ROM:C0018990 BL sub_C006B354
ROM:C0018994 MOV R3, #1
ROM:C0018998 STR R3, [R5]

```

data cross-references  
 code cross-references

(b) The image base is set to 0xC0018000

FIGURE 1: Comparison of incorrect and correct image base disassembly results.

other devices. The firmware can be obtained by downloading it from the website of the vendor or extracting it from the flash storage of the device hardware. Any firmware used in smart devices should be assumed insecure, which may have security vulnerabilities.

To evaluate and improve the security of firmware, a necessary method is disassembling [9, 10]. In this case, a disassembler, such as IDA Pro, needs to know the processor

type and image base of the firmware [11]. In general, the processor type can be discerned by consulting the product manual or physical examination of the hardware [12, 13]. However, the image base cannot be obtained directly. Without the image base, the disassembler is unable to create cross-references based on absolute addresses [14]. When these cross-references are lacking, it is difficult to navigate efficiently in disassembly listing. Facing the obscure disassembly

code, people often lose their direction when they look for the assembly code in which they are most interested. Conversely, knowledge of the correct image base is critical in understanding the firmware as a whole [12].

Heterogeneous hardware architectures are used in firmware images; however, many smart devices are based on the ARM architecture [15–17]. Therefore, this work mainly focuses on ARM-based firmware. As shown in Figure 1, Figure 1(a) shows the disassembly code with the wrong image base and Figure 1(b) shows the disassembly code with the correct image base. IDA Pro cannot establish a cross-reference when the wrong image base is set, and the absolute addresses are marked in red. When the correct image base is set, IDA Pro establishes cross-references to these absolute addresses, which are important for reverse engineers to understand the intention of the assembly code.

To determine the image base of firmware, many researchers have put in a great deal of effort, and several manual solutions have been proposed.

Skochinsky [18] proposed a general principle for determining the image base of a file with an unknown format. He suggested that some kinds of hints, such as self-relocating code and initialization code, can be used.

Basnight et al. [12, 19] presented two methods for inferring the image base. The first method uses immediate values in instruction to infer a reasonable image base. The second method uses a hardware debugger to halt a programmable logic controller and obtain a memory dump. Then, the image base can be found by manually analyzing common instruction patterns in the memory dump.

Dacosta et al. [20] noted that when the case values in a switch-case statement of a C program are sequential and dense, the memory addresses of the case are usually stored in a jump table; this fact can be used to infer the memory address of the nearby code and eventually obtain the image base. Dacosta's approach manually analyzed the instruction of jump to default statement block (in this case, the BHI instruction) first, obtained the offset of the default statement block, and then analyzed the memory address of the default statement block to calculate the image base.

All of the above methods are not automated and heavily rely on reverse engineers' experience and intuition. We have proposed [21–23] three methods for automatically determining the image base. These automated methods are applicable to different types of ARM firmware, which cannot determine the image base of all types of firmware.

In this paper, we proposed a method for determining the image base of firmware that uses a jump table to store absolute addresses. The source code of firmware usually contains switch-case statements, and the compiler may generate jump tables for such code. By searching the sequence of instructions, the jump table can be located. Then, according to the absolute addresses in the jump table and the offset of the case statement block, we can obtain the image base. The experimental result indicates that the proposed method can effectively determine the image base of firmware that uses the jump table to store the absolute addresses.

```

switch(n)
{
  case 0:
    printf("n =0\n");
    break;
  case 1:
    printf("n =1\n");
    break;
  case 2:
    printf("n =2\n");
    break;
  case 3:
    printf("n =3\n");
    break;
  case 4:
    printf("n =4\n");
    break;
  default:
    printf("default.\n");
}

```

LISTING 1: Example of switch-case statements.

## 2. Jump Table in Firmware

The switch-case statement often appears in the source code of firmware and may generate a jump table after being compiled. After the code in Listing 1 is compiled into a binary file, IDA Pro can be used to disassemble the binary file, and the disassembly results are shown in Figure 2.

It can be seen that when there is a switch-case statement in the code, the compiler may generate a jump table. The content in the jump table is the addresses of the case statement block; for example, 0x8268 in the jump table is the address of the first case statement block.

Next, we analyze the calculation process of the jump table in two cases.

- (1) Suppose that variable  $n$  in the code of Listing 1 is less than or equal to 4 (e.g., 3), then register R3 in the instruction at memory 0x8248 in Figure 2 is 3. After executing the instruction "CMP R3, #4" at offset 0x00008248, the LDRLS instruction is executed. According to the ARM manual [24], the memory address accessed by LDRLS is

$$\begin{aligned}
 \text{address} &= \text{PC} + (\text{R3} * 4) \\
 &= (\text{Current} + 8) + (\text{R3} * 4) \\
 &= (0x824C + 8) + (0x3 * 4) \\
 &= 0x8260.
 \end{aligned} \tag{1}$$

As shown in Figure 2, the word at address 0x8260 is 0x828C. This means that the PC register will be assigned a value of 0x828C, and the program will jump to 0x828C to continue execution

```

.text:00008248      CMP     R3, #4
.text:0000824C      LDRLS  PC, [PC,R3,LSL#2]
.text:00008250      B      loc_82A4
.text:00008250 ;
.text:00008254      DCD    0x8268      Jump Table
.text:00008258      DCD    0x8274
.text:0000825C      DCD    0x8280
.text:00008260      DCD    0x828C
.text:00008264      DCD    0x8298
.text:00008268 ;
.text:00008268
.text:00008268  loc_8268
.text:00008268      LDR     R0, =aN0
.text:0000826C      BL      puts
.text:00008270      B      loc_82AC
.text:00008274 ;
.text:00008274
.text:00008274  loc_8274
.text:00008274      LDR     R0, =aN1
.text:00008278      BL      puts
.text:0000827C      B      loc_82AC
.text:00008280 ;
.text:00008280
.text:00008280  loc_8280
.text:00008280      LDR     R0, =aN2
.text:00008284      BL      puts
.text:00008288      B      loc_82AC
.text:0000828C ;
.text:0000828C
.text:0000828C  loc_828C
.text:0000828C      LDR     R0, =aN3
.text:00008290      BL      puts
.text:00008294      B      loc_82AC
.text:00008298 ;
.text:00008298
.text:00008298  loc_8298
.text:00008298      LDR     R0, =aN4
.text:0000829C      BL      puts
.text:000082A0      B      loc_82AC
.text:000082A4 ;
.text:000082A4
.text:000082A4  loc_82A4
.text:000082A4      LDR     R0, =aDefault_
.text:000082A8      BL      puts
.text:000082AC
.text:000082AC  loc_82AC
.text:000082AC
.text:000082AC      MOV     R3, #0
.text:000082B0      MOV     R0, R3
.text:000082B4      SUB     SP, R11, #4
.text:000082B8      LDMFD  SP!, {R11,LR}
.text:000082BC      BX     LR

```

FIGURE 2: Disassembly code.

- (2) When the value of variable  $n$  is greater than 4, i.e., the value of  $R3$  is greater than 4, the instruction “B loc\_82A4” at offset 0x00008250 will be executed. The program will jump to location loc\_82A4 to continue execution

According to the above analysis, we can understand the calculation process of the jump table. Take the firmware of ABB NETA-21 as a case, as shown in Figure 3. The CMP instruction at offset 0x000AB124 is followed by the LDRLS

instruction, the B instruction, and a jump table. The jump table begins at offset 0x000AB130 with four addresses, as shown by the red background in Figure 3, which are 0xC00B326C, 0xC00B3160, 0xC00B3150, and 0xC00B3140. In general, the minimum memory address in the jump table points to the first case statement block, and the first case statement block is usually next to the jump table. The minimum memory address in the jump table is 0xC00B3140, and the first case statement block starts at offset 0x000AB140. That is, the case statement block with offset 0x000AB140 is

ROM:000AB118	MOV	R0, R5	
ROM:000AB11C	LDR	R1, [SP,#0x60+var_2C]	
ROM:000AB120	BL	sub_1E0A90	
ROM:000AB124	CMP	R4, #3 ; switch 4 cases	
ROM:000AB128	LDRLS	PC, [PC,R4,LSL#2] ; switch jump	
ROM:000AB12C	B	loc_AB264 ; jumptable 000AB128 default case	
ROM:000AB130	DCD	0xC00B326C	Jump Table
ROM:000AB130	DCD	0xC00B3160	
ROM:000AB130	DCD	0xC00B3150	
ROM:000AB130	DCD	0xC00B3140	
ROM:000AB140	LDR	R3, =0xC00B3344	
ROM:000AB144	LDR	R8, [R7,#0x128]	
ROM:000AB148	STR	R3, [R7,#0x118]	
ROM:000AB14C	B	loc_AAF7C	
ROM:000AB150	LDR	R3, =0xC00B336C	
ROM:000AB154	LDR	R8, [R7,#0x128]	
ROM:000AB158	STR	R3, [R7,#0x118]	
ROM:000AB15C	B	loc_AAF7C	
ROM:000AB160	LDR	R3, =0xC00B3388	
ROM:000AB164	LDR	R8, [R7,#0x128]	
ROM:000AB168	STR	R3, [R7,#0x118]	
ROM:000AB16C	B	loc_AAF7C	
ROM:000AB170			

FIGURE 3: Jump table in ABB NETA-21 firmware uImage (the image base is set to 0).

mapped to the memory address 0xC00B3140, and then, the image base can be calculated.

### 3. DBJT Algorithm

According to the above analysis, when compiling the switch-case statement, the compiler usually generates the CMP instruction, LDRLS instruction, B instruction, and jump table in turn. The program jumps according to the addresses in the jump table. The model is shown in Figure 4.

In general, the minimum memory address in the jump table points to the first case statement block. A jump table can be used to deduce the memory address of the first case statement block; thus, the difference between the memory address and offset of the first case statement block can be used to obtain the candidate image base.

Figure 5 shows that the firmware that contains a case block with offset  $offset\_case1$  is mapped to memory. The image base of firmware is denoted as the  $base$ , and the minimum memory address in the jump table is denoted as  $min\_addr$ . According to the analysis in Section 2, the first case block with offset  $offset\_case1$  is mapped to memory location  $min\_addr$ , i.e.,  $min\_addr = base + offset\_case1$ , and then, we can obtain the image base as  $base = min\_addr - offset\_case1$ .

Based on the model of the switch-case statement, we can scan from the starting position of the firmware to locate the switch-case statement. If in a location, the current instruction is CMP, the second instruction is LDRLS, and the third instruction is B, then we consider it to be a switch-case statement, and the B instruction is followed by

the jump table. Then, read in all the content of the jump table, obtain the minimum element of the jump table, and subtract the offset of the first case block from the minimum element to obtain a candidate image base. With one jump table, we can obtain a candidate image base. All candidate image bases can be calculated from all jump tables of the firmware. Then, we count the frequency of each candidate image base. If the frequency of a particular candidate image base is much larger than those of others, then we consider this candidate to be the actual image base. Based on the above analysis, we propose the determining the image base by searching jump tables (DBJT) algorithm to determine the image base. The pseudocode of the algorithm is shown in Listing 2.

The time complexity of the DBJT algorithm is  $O(fileSize)$ , where  $fileSize$  is the size of the firmware file. The algorithm first locates the jump table according to three consecutive instructions (CMP instruction, LDRLS instruction, and B instruction) and then sorts all the addresses in the jump table to obtain the minimum memory address. A candidate image base is obtained by the difference between the offset of the case statement block and the minimum memory address, and the candidate image base is added to multiset  $M$ . Finally, count the number of occurrences of each candidate image base in the multiset  $M$ , and then, sort them in descending order by occurrences. If a candidate image base appears much more frequently than other elements, then it is considered the correct image base. Otherwise, the outputs do not contain the correct image base because the DBJT algorithm cannot be applied successfully to this firmware.

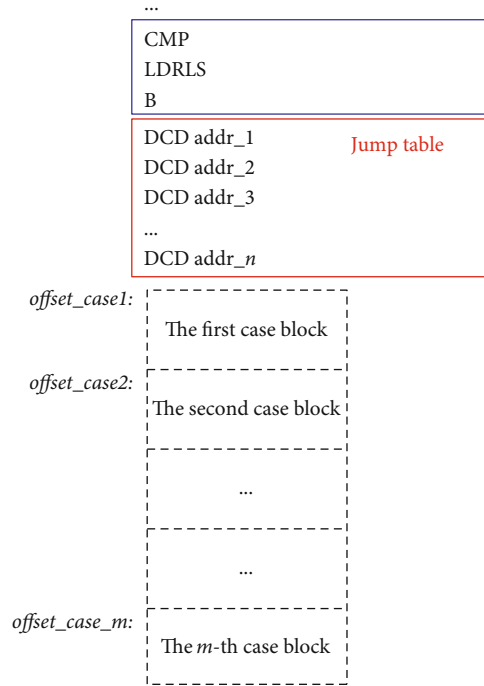


FIGURE 4: The assembly model of the switch-case statement.

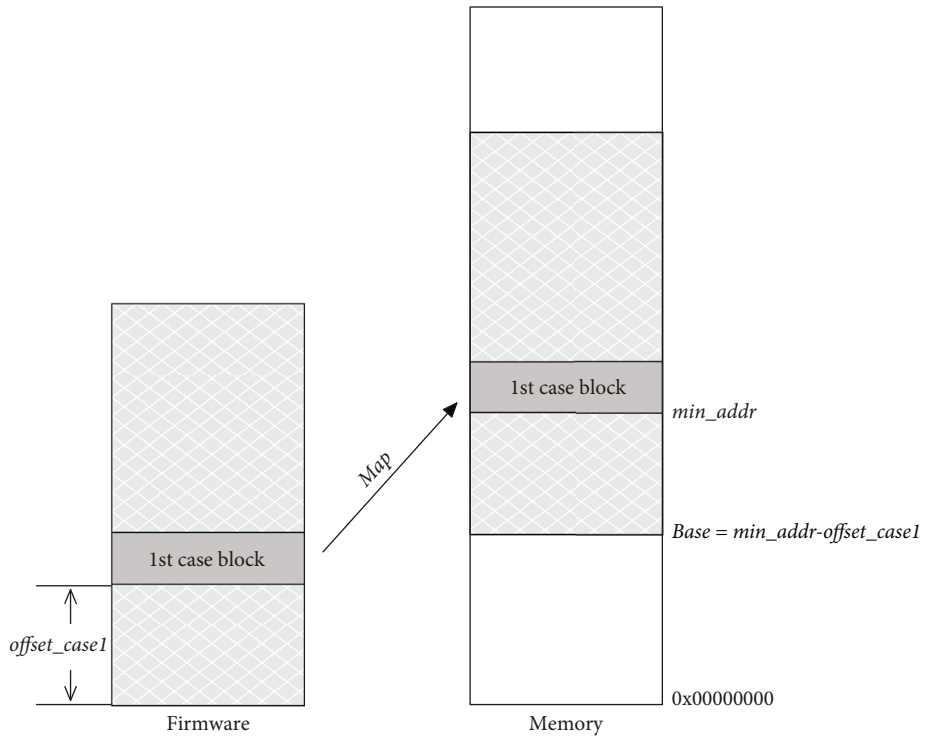


FIGURE 5: Map firmware into memory.

### 4. Experimental Results and Analysis

To test the proposed algorithm, we collected 10 firmware from well-known vendors’ official websites. The DBJT algorithm was implemented in the C language and was compiled with Visual C++6.0. The experiments were performed on a

personal computer with an Intel i7-2600 3.4 GHz processor and 18 GB memory, running Microsoft Windows 7 SP1.

4.1. *Experimental Results.* In the experiment, the DBJT algorithm proposed in this paper is used to identify the jump table in the firmware and calculate the image base. The

```

Input: firmwareFile
Output: A sorted result of the elements and their occurrence in multiset M
function DBJT (firmwareFile)
  fileSize  $\leftarrow$  Obtain the size of firmwareFile
  offset  $\leftarrow$  0
  while( $0 \leq \textit{offset} < \textit{fileSize}$ ) do
    CMP_FLAG  $\leftarrow$  FALSE
    LDRLS_FLAG  $\leftarrow$  FALSE
    B_FLAG  $\leftarrow$  FALSE
    if Current instruction is CMP instruction, then
      CMP_FLAG  $\leftarrow$  TRUE
    else
      offset  $\leftarrow$  offset + 4
      continue
    end if
    if The second instruction is LDRLS instruction, then
      LDRLS_FLAG  $\leftarrow$  TRUE
    else
      offset  $\leftarrow$  offset + 4
      continue
    end if
    if The third instruction is B instruction, then
      B_FLAG  $\leftarrow$  TRUE
    else
      offset  $\leftarrow$  offset + 4
      continue
    end if
    if CMP_FLAG == TRUE && LDRLS_FLAG == TRUE && B_FLAG == TRUE then
      jt[n]  $\leftarrow$  Read the jump table
      min_addr  $\leftarrow$  Obtain the minimum element of the array jt[n]
      offset_case1  $\leftarrow$  Obtain offset of the first case block
      base  $\leftarrow$  min_addr - offset_case1
      if base % 4 == 0 then
        M  $\leftarrow$  base
      end if
      offset  $\leftarrow$  offset_case1
    end if
    offset  $\leftarrow$  offset + 4
  end while
  Count the number of occurrences of each element in the multiset M
  Sort the elements and their occurrence in descending order by number of occurrences
  Output: Sorted elements and their occurrences
end function

```

LISTING 2: Determining the image base by searching jump tables (DBJT).

TABLE 1: Experimental results of the DBJT algorithm.

Device	Firmware	Jump table	Correct	Base	Time (ms)	Validated
ABB NETA-21	uImage	261	108	0xC0008000	250	Yes
Advantech 4570-CE	57791ec9.bin	222	38	0x7F000000	172	Yes
Advantech 2748FI Switch	3551.bin	279	272	0x00400000	93	Yes
Emerson ES-03001	es-03001-1.ffd	0	0	N/A	31	N/A
Phoenix 400 PND-4TX-IB	2985563_321.fw	448	437	0x20800F28	546	Yes
Phoenix OT 4 M Terminal	v1.23.nb0	0	0	N/A	15	N/A
Rockwell DriveLogix 5730	pn-82672.bin	0	0	N/A	47	N/A
Schneider 140CRA31200	cra31200.bin	318	153	0x00001000	156	Yes
Schneider 140CRA31200	140cra31200.bin	217	111	0x02001000	109	Yes
Schneider M241 PLC	vxBoot.bin	43	20	0x00801FC0	93	Yes

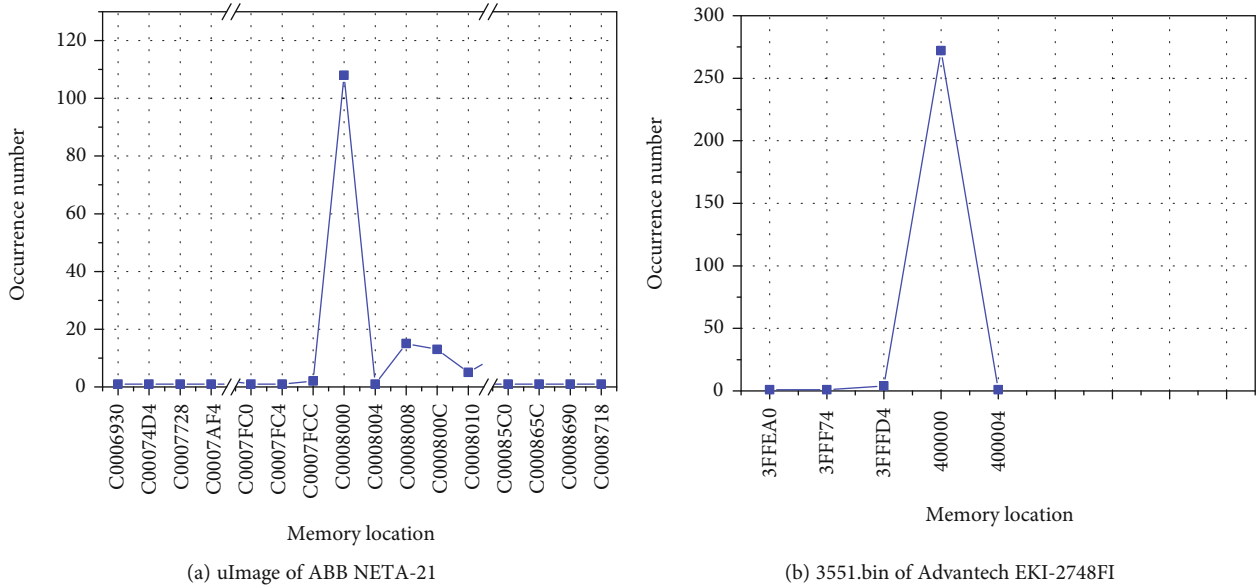


FIGURE 6: Image base determination results.

```

ROM:C00A06C4      CMP     R3, #0xA
ROM:C00A06C8      LDRLS  PC, [PC,R3,LSL#2]
ROM:C00A06CC      B       loc_C00A0778
ROM:C00A06CC ; -----
ROM:C00A06D0      DCD   loc_C00A0770
ROM:C00A06D0      DCD   loc_C00A0764
ROM:C00A06D0      DCD   loc_C00A0758
ROM:C00A06D0      DCD   loc_C00A074C
ROM:C00A06D0      DCD   loc_C00A0744
ROM:C00A06D0      DCD   loc_C00A0738
ROM:C00A06D0      DCD   loc_C00A072C
ROM:C00A06D0      DCD   loc_C00A0720
ROM:C00A06D0      DCD   loc_C00A0714
ROM:C00A06D0      DCD   loc_C00A0708
ROM:C00A06D0      DCD   loc_C00A06FC
ROM:C00A06FC ; -----
ROM:C00A06FC      loc_C00A06FC
ROM:C00A06FC      LDRB  R3, [R12,#0xA]
ROM:C00A0700      MOV   R3, R3,LSL#24
ROM:C00A0704      ADD   R0, R0, R3
ROM:C00A0708
ROM:C00A0708      loc_C00A0708
ROM:C00A0708      LDRB  R3, [R12,#9]
ROM:C00A070C      MOV   R3, R3,LSL#16
ROM:C00A0710      ADD   R0, R0, R3
ROM:C00A0714
ROM:C00A0714      loc_C00A0714
ROM:C00A0714      LDRB  R3, [R12,#8]
    
```

(a) The image base is set to 0xC0080000

```

ROM:000986C4      CMP     R3, #0xA
ROM:000986C8      LDRLS  PC, [PC,R3,LSL#2]
ROM:000986CC      B       loc_98778
ROM:000986CC ; -----
ROM:000986D0      DCD   0xC00A0770
ROM:000986D0      DCD   0xC00A0764
ROM:000986D0      DCD   0xC00A0758
ROM:000986D0      DCD   0xC00A074C
ROM:000986D0      DCD   0xC00A0744
ROM:000986D0      DCD   0xC00A0738
ROM:000986D0      DCD   0xC00A072C
ROM:000986D0      DCD   0xC00A0720
ROM:000986D0      DCD   0xC00A0714
ROM:000986D0      DCD   0xC00A0708
ROM:000986D0      DCD   0xC00A06FC
ROM:000986FC ; -----
ROM:000986FC      LDRB  R3, [R12,#0xA]
ROM:00098700      MOV   R3, R3,LSL#24
ROM:00098704      ADD   R0, R0, R3
ROM:00098708      LDRB  R3, [R12,#9]
ROM:0009870C      MOV   R3, R3,LSL#16
ROM:00098710      ADD   R0, R0, R3
ROM:00098714      LDRB  R3, [R12,#8]
    
```

(b) The image base is set to 0

FIGURE 7: The disassembly result of the correct and incorrect image base.



experimental results are shown in Table 1. The column “Jump table” lists the number of jump tables identified by the DBJT algorithm in each firmware file. The column “Correct” lists the frequency of the correct image base identified by the DBJT algorithm, and the column “Base” lists the correct image bases of the corresponding firmware. The column “Time” lists the execution time of the proposed algorithm. The symbol N/A means that the method is not applicable to the corresponding firmware; the reasons for this are discussed in Section 4.2. The manual validation results are shown in the “Validated” column of Table 1.

We take the firmware uImage of ABB NETA-21 as an example to analyze the experimental results. As shown in Table 1, 261 jump tables are identified by the DBJT algorithm, 108 of which point to the same candidate image base 0xC0008000. Figure 6(a) shows the candidate image base and the corresponding occurrence frequency. It can be seen that the candidate image base 0xC0008000 appears 108 times, which is much higher than the frequency of other candidate image bases. The practical significance is that the candidate image base calculated by 108 jump tables is 0xC0018000. Therefore, we consider 0xC0018000 to be the correct image base of the firmware.

To verify whether the experimental results are correct, we load the firmware file uImage using IDA Pro and set the processor type to “ARM little-endian” and the image base to 0xC0008000. Then, we can see that the cross-references for absolute addresses in the disassembly code are correct, as shown in Figure 7(a). This indicates that the memory address 0xC0008000 is the correct image base. In comparison, the same file loaded by IDA Pro without setting the correct image base is shown in Figure 7(b).

As shown in Table 1, the execution time of the proposed algorithm for uImage is 250 ms. Compared to the time of reverse engineering, the time to determine the image base is insignificant.

Figure 6(b) shows the experimental results obtained for the firmware sample 3551.bin from the Advantech EKI-2748FI-managed Ethernet switch, the image base of which is 0x00400000, which is manually verified as the correct image base.

In Figure 6, we can see that there are some other points near the image base. These points are caused by errors in the algorithm. If the default statement block is in the first position in the switch-case statement, then the minimum memory address in the jump table no longer points to the first case statement block, and the default statement block is next to the jump table. This style of the C code is shown in Listing 3, and its corresponding assembly code is shown in Figure 8. Although such style of the C code is legitimate, most programmers never write in such style. This type of switch-case statement will lead to the inaccuracy of the DBJT algorithm, which will differ from the correct image base by a few bytes.

**4.2. Possible Reasons for Determination Failure.** In Table 1, the number of recognized jump tables in some firmware is 0, and the image base is not determined successfully, indicating that the DBJT algorithm is not suitable for this firmware. The possible reasons for this are as follows.

```

switch(n)
{
  default:
    printf("default.\n");
    break;
  case 0:
    printf("n =0\n");
    break;
  case 1:
    printf("n =1\n");
    break;
  case 2:
    printf("n =2\n");
    break;
  case 3:
    printf("n =3\n");
    break;
  case 4:
    printf("n =4\n");
    break;
}

```

LISTING 3: Example of switch-case statements.

- (1) The compiler generates a jump table only when the value of the case in the switch-case is sequential and dense. Otherwise, the compiler generates no jump table. For example, the case value in Listing 4 is not sequential, and there is no jump table generated, as shown in Figure 9
- (2) In some firmware, the jump table contains no absolute addresses, and the DBJT algorithm cannot be used to determine the image base, such as firmware es-03001-1.fdd of Emerson ES-03001, firmware v1.23.nb0 of Phoenix OT 4M Terminal, and pn-82672.bin of Rockwell DriveLogix 5730. Figure 10 shows the assembly code of firmware es-03001-1.fdd

In Figure 10, the BHI instruction at address 0x00004E00 is the “Branch if Higher” instruction. Combined with the previous instruction, “CMP R1, #6,” if R1 is greater than 6, then it will jump to the label def\_4E0C. If R1 is less than or equal to 6 (e.g., 2), then the ADR instruction will be executed. The ADR instruction at address 0x00004E04 assigns register R2 to 0x00004E10. LDRB instruction loads a byte from memory. Then,  $R2 + R1 = 0x00004E10 + 0x2 = 0x00004E12$ . The 0x01 at address 0x00004E12 is loaded into register R2. The ADD instruction at address 0x00004E0C will modify the value of the PC register. The calculation process of the PC register is as follows:

$$\begin{aligned}
 PC &= PC + R2 * 4 \\
 &= (\text{Current} + 8) + (R2 * 4) \\
 &= (0x4E0C + 8) + (0x01 * 4) \\
 &= 0x4E18.
 \end{aligned} \tag{2}$$

```

    .text:00008248          CMP     R3, #4
    .text:0000824C          LDRLS  PC, [PC,R3,LSL#2]
    .text:00008250          B      loc_8268
    .text:00008250 ; -----
    .text:00008254          DCD    0x8274
    .text:00008258          DCD    0x8280
    .text:0000825C          DCD    0x828C
    .text:00008260          DCD    0x8298
    .text:00008264          DCD    0x82A4
    .text:00008268 ; -----
    .text:00008268
    .text:00008268 loc_8268
    .text:00008268          LDR     R0, =aDefault_
    .text:0000826C          BL     puts
    .text:00008270          B      loc_82AC
    .text:00008274 ; -----
    .text:00008274
    .text:00008274 loc_8274
    .text:00008274          LDR     R0, =aN0
    .text:00008278          BL     puts
    .text:0000827C          B      loc_82AC
    .text:00008280 ; -----
    .text:00008280
    .text:00008280 loc_8280
    .text:00008280          LDR     R0, =aN1
    .text:00008284          BL     puts
    .text:00008288          B      loc_82AC
    .text:0000828C ; -----
    .text:0000828C
    .text:0000828C loc_828C
    .text:0000828C          LDR     R0, =aN2
    .text:00008290          BL     puts
    .text:00008294          B      loc_82AC
    .text:00008298 ; -----
    .text:00008298
    .text:00008298 loc_8298
    .text:00008298          LDR     R0, =aN3
    .text:0000829C          BL     puts
    .text:000082A0          B      loc_82AC
    .text:000082A4 ; -----

```

FIGURE 8: Disassembly code.

```

switch(n)
{
case 1:
    printf("n =1\n");
    break;
case 100:
    printf("n =100\n");
    break;
default:
    printf("default.\n");
}

```

LISTING 4: Example of switch-case statements.

That is, the PC register will be assigned the value 0x4E18. From the above calculation, it can be seen that there is no absolute address stored in the jump table, so the algorithm proposed in this paper cannot be used for this firmware.

## 5. Conclusions

The disassembly of firmware is a necessary step in the security assessment of authentication mechanisms. However, for the firmware of most smart devices, the image base cannot be obtained directly, which is a major obstacle to disassembly. In this paper, we research the storage law of the jump table in the ARM firmware of smart devices and

```

• .text:00008240          STR     R3, [R11,#var_8]
• .text:00008244          LDR     R3, [R11,#var_8]
• .text:00008248          STR     R3, [R11,#var_18]
• .text:0000824C          LDR     R3, [R11,#var_18]
• .text:00008250          CMP     R3, #1
• .text:00008254          BEQ     loc_8268
• .text:00008258          LDR     R3, [R11,#var_18]
• .text:0000825C          CMP     R3, #0x64
• .text:00008260          BEQ     loc_8274
• .text:00008264          B       loc_8280
; -----
• .text:00008268          ;
• .text:00008268          loc_8268
• .text:00008268          LDR     R0, =aN1
• .text:0000826C          BL      puts
• .text:00008270          B       loc_8288
; -----
• .text:00008274          ;
• .text:00008274          loc_8274
• .text:00008274          LDR     R0, =aN100
• .text:00008278          BL      puts
• .text:0000827C          B       loc_8288
; -----
• .text:00008280          ;
• .text:00008280          loc_8280
• .text:00008280          LDR     R0, =aDefault_
• .text:00008284          BL      puts
• .text:00008288          loc_8288
• .text:00008288          ;
• .text:00008288          MOV     R3, #0

```

FIGURE 9: Disassembly code.

```

ROM:00004DFC          CMP     R1, #6
ROM:00004E00          BHI     def_4E0C
ROM:00004E04          ADR     R2, jpt_4E0C
ROM:00004E08          LDRB    R2, [R2,R1]
ROM:00004E0C          ADD     PC, PC, R2,LSL#2
ROM:00004E0C          ; -----
ROM:00004E10          jpt_4E0C          DCB  0x22 ; "          Jump Table
ROM:00004E11          DCB  0x22 ; "
ROM:00004E12          DCB  1
ROM:00004E13          DCB  6
ROM:00004E14          DCB  0xF
ROM:00004E15          DCB  0x1B
ROM:00004E16          DCB  0x20
ROM:00004E17          DCB  0
ROM:00004E18          ; -----
ROM:00004E18          loc_4E18
ROM:00004E18          LDR     R1, =0x25AB02
ROM:00004E1C          LDRB    R0, [R1]
ROM:00004E20          CMP     R0, #2
ROM:00004E24          BNE     def_4E0C
ROM:00004E28          B       loc_4E74
; -----
ROM:00004E2C          ;
ROM:00004E2C          loc_4E2C
ROM:00004E2C          LDRB    R0, [R4,#0x247]
ROM:00004E30          CMP     R0, #1
ROM:00004E34          BEQ     loc_4E58
ROM:00004E38          LDR     R0, [R4,#0x278]
ROM:00004E3C          BL      sub_6CEF4

```

FIGURE 10: Jump table in Emerson ES-03001 firmware es-03001-1.ffd (the image base is set to 0).

propose a method for determining the firmware image base by using a jump table. The experimental results show that the proposed method is effective for the firmware that stores the absolute addresses in the jump table. For future work, it is still a challenge to automatically determine the image base of other types of firmware, such as firmware that contains no jump table. We will continue to research new methods for other kinds of firmware in smart devices. We believe that these automated approaches can effectively reduce the difficulty of reverse analysis.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61802439) and Beijing Youth Backbone Personal Project (Grant No. 201800002685XG357).

## References

- [1] "IoT connections outlook|Mobility report - Ericsson," <https://www.ericsson.com/en/mobility-report/reports/november-2019/iot-connections-outlook>.
- [2] "Reverse engineering a D-Link backdoor," <http://www.devttys0.com/2013/10/reverse-engineering-a-d-link-backdoor/>.
- [3] "From China, with love," <http://www.devttys0.com/2013/10/from-china-with-love/>.
- [4] W. Wang, X. Wang, D. Feng, J. Liu, Z. Han, and X. Zhang, "Exploring permission-induced risk in Android applications for malicious application detection," *s*, vol. 9, no. 11, pp. 1869–1882, 2014.
- [5] W. Li, W. Meng, Z. Tan, and Y. Xiang, "Design of multi-view based email classification for IoT systems via semi-supervised learning," *Journal of Network and Computer Applications*, vol. 128, pp. 56–63, 2019.
- [6] W. Wang, Y. Shang, Y. He, Y. Li, and J. Liu, "BotMark: automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors," *Information Sciences*, vol. 511, pp. 284–296, 2020.
- [7] W. Meng, W. Li, and L. Kwok, "EFM: enhancing the performance of signature-based network intrusion detection systems using enhanced filter mechanism," *Computers & Security*, vol. 43, pp. 189–204, 2014.
- [8] Z. Guan, X. Liu, L. Wu et al., "Cross-lingual multi-keyword rank search with semantic extension over encrypted data," *Information Sciences*, vol. 514, pp. 523–540, 2020.
- [9] L. Zhang, S. Hao, J. Zheng, Y. Tan, Q. Zhang, and Y. Li, "Descrambling data on solid-state disks by reverse-engineering the firmware," *Digital Investigation*, vol. 12, pp. 77–87, 2015.
- [10] Z. Liu, Y. Huang, J. Li, X. Cheng, and C. Shen, "DivORAM: towards a practical oblivious RAM with variable block size," *Information Sciences*, vol. 447, pp. 1–11, 2018.
- [11] P. Shirani, L. Collard, B. L. Agba et al., "BINARM: scalable and efficient detection of vulnerabilities in firmware images of intelligent electronic devices," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, 2018.
- [12] Z. Basnight, J. Butts, J. Lopez, and T. Dube, "Firmware modification attacks on programmable logic controllers," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 2, pp. 76–84, 2013.
- [13] J. C. Mulder, M. D. Schwartz, M. J. Berg, J. R. Van Houten, J. M. Urrea, and A. N. Pease, "Reverse engineering industrial control system field devices," in *International Conference on Critical Infrastructure Protection*, Albuquerque, NM, USA, 2012.
- [14] C. D. Schuett, *Programmable logic controller modification attacks for use in detection analysis*, DTIC Document, 2014.
- [15] B. Chen, X. Dong, G. Bai, S. Jauhar, and Y. Cheng, "Secure and efficient software-based attestation for industrial control devices with arm processors," in *Proceedings of the 33rd Annual Computer Security Applications Conference*, New York, NY, USA, 2017.
- [16] Y. J. Kwon, H. K. Kim, K. M. Koumadi, Y. H. Lim, and J. In Lim, "Automated vulnerability analysis technique for smart grid infrastructure," in *IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC, USA, 2017.
- [17] A. Costin, J. Zaddach, A. Francillon, D. Balzarotti, and Eurecom, "A large-scale analysis of the security of embedded firmwares," in *Proceedings of the 23rd USENIX conference on Security Symposium*, San Diego, CA, 2014.
- [18] I. Skochinsky, "Intro to embedded reverse engineering for PC reversers," in *REcon Conference*, Montreal, Canada, 2010.
- [19] Z. H. Basnight, *Firmware counterfeiting and modification attacks on programmable logic controllers*, Air Force Institute of Technology, Ohio, 2013.
- [20] I. Dacosta, N. Mehta, E. Metrock, and J. Giffin, "Security analysis of an IP phone: Cisco 7960G," in *Principles, Systems and Applications of IP Telecommunications. Services and Security for Next Generation Networks*, Springer-Verlag, 2008.
- [21] R. Zhu, Y. Tan, Q. Zhang, Y. Li, and J. Zheng, "Determining image base of firmware for ARM devices by matching literal pools," *Digital Investigation*, vol. 16, pp. 19–28, 2016.
- [22] R. Zhu, Y. Tan, Q. Zhang, W. Fei, J. Zheng, and Y. Xue, "Determining image base of firmware files for ARM devices," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 2, pp. 351–359, 2016.
- [23] R. Zhu, B. Zhang, J. Mao, Q. Zhang, and Y. Tan, "A methodology for determining the image base of ARM-based industrial control system firmware," *International Journal of Critical Infrastructure Protection*, vol. 16, pp. 26–35, 2017.
- [24] ARM Limited, *ARM Architecture Reference Manual*, ARM Limited, 2014.

## Research Article

# Study on Security and Privacy in 5G-Enabled Applications

Qin Qiu <sup>1</sup>, Shenglan Liu <sup>2</sup>, Sijia Xu <sup>1</sup>, and Shengquan Yu <sup>3</sup>

<sup>1</sup>China Mobile Communications Group Co., Ltd., Beijing 100053, China

<sup>2</sup>China Mobile Group Design Institute Co., Ltd., Beijing 100080, China

<sup>3</sup>Advanced Innovation Center for Future Education, Beijing Normal University, Beijing 102206, China

Correspondence should be addressed to Shengquan Yu; [yusq@bnu.edu.cn](mailto:yusq@bnu.edu.cn)

Received 11 August 2020; Revised 22 November 2020; Accepted 7 December 2020; Published 21 December 2020

Academic Editor: Ding Wang

Copyright © 2020 Qin Qiu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

5G applications face security risks due to the new technology used and the performance requirements of the specific application scenario. This paper analyzes the security requirements and presents hierarchical solutions for stakeholders to build secure 5G applications. First, we summarize the technical characteristics and typical usage scenarios of 5G. Then, we analyze the security and privacy risks faced by 5G applications and related security standards and research work. Next, we give the system reference architecture and overall security and privacy solutions for 5G applications. Based on the three major application scenarios of eMBB, uRLLC, and mMTC, we also provide specific suggestions for coping with security and privacy risks. Finally, we present a use case of industrial terminal access control and make conclusions of this paper.

## 1. Introduction

The fifth-generation mobile network (5G) is a new generation mobile network that enables innovations and progressive changes across all vertical industries like smart grids and smart campus [1]. 5G mobile communication technology is based on a new architecture [2]. The 3rd Generation Partnership Project (3GPP) has provided complete system specifications for 5G network architecture (see Figure 1). Components of the core network can be instantiated multiple times to support virtualization technologies and network slicing. The architecture is driven by the motivation to remove the data overlay that has been traditionally used in previous generations of mobile networks [3].

The introduction of new key technologies such as network function virtualization (NFV), software-defined network (SDN), network slicing, multiaccess edge computing (MEC) [5], mm-Wave communication [6], and massive MIMO [7] greatly improves the network's support for various applications. The International Telecommunication Union (ITU) identifies three new usage scenarios of 5G (depicted in Figure 2), which are enhanced mobile broadband (eMBB), ultrareliable and low latency communications

(uRLLC), and massive machine type communications (mMTC), and proposes eight key performance indicators (KPI) [7]. Regarding these KPIs, 5G has high performances, reaching 10 times the peak rate of 4G, shortening the transmission latency to milliseconds, and handling a million concurrent connections per square kilometer [8, 9]. The rich and diverse 5G applications and their broad development prospects initiate a new era of ubiquitous and intelligent internet. The European Union even predicts that 5G will become the backbone of vital societal and economic functions—such as energy, transport, banking, and health, as well as industrial control systems [10]. According to HIS Markit [11], 5G will generate a global economic output worth \$13.2 trillion and create 22.3 million jobs by 2035.

As 5G new technology and the performance requirements of specific application scenarios bring about many security risks, security has become a priority when stakeholders develop 5G vertical applications. This paper makes contributions in the following aspects:

- (1) Analyzes the technical characteristics of 5G technologies and use cases of 5G applications. Then summarizes typical vertical applications enabled by 5G technologies,

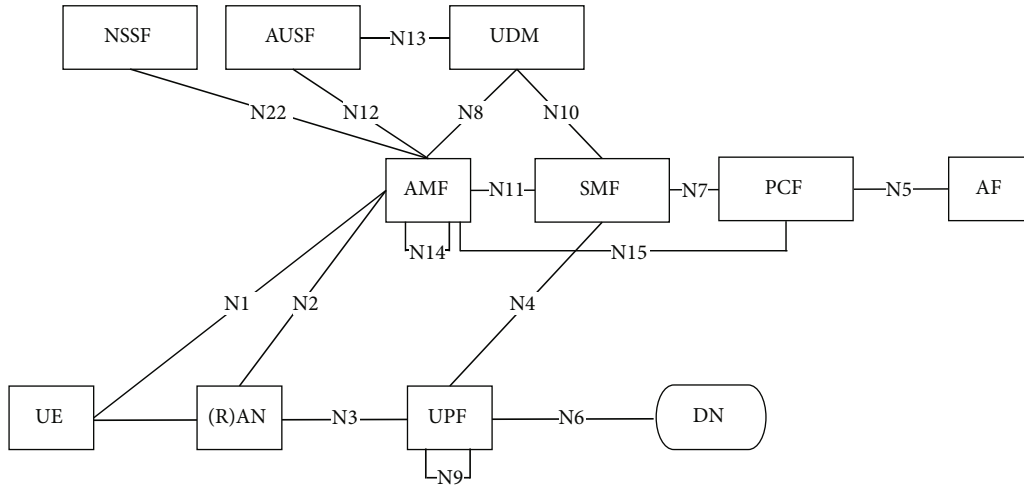


FIGURE 1: 3GPP 5G system architecture for nonroaming cases [4].

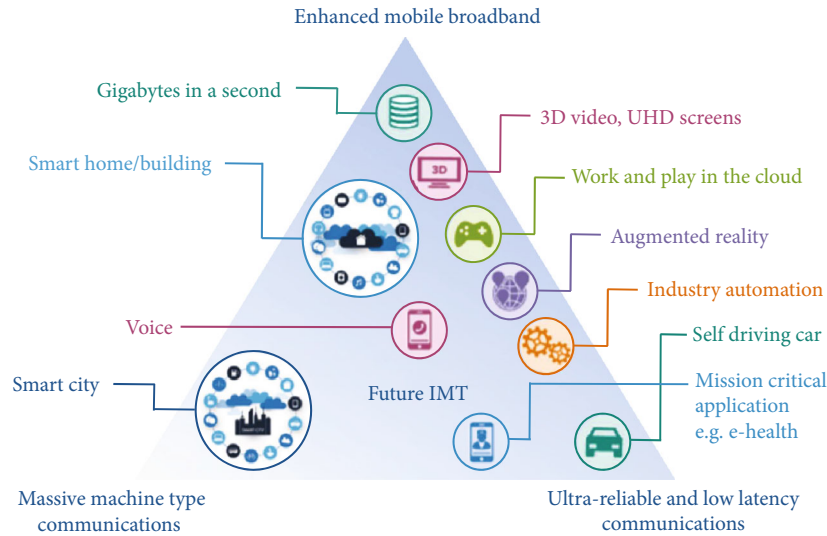


FIGURE 2: 5G main usage scenarios defined by ITU [8].

- involving smart manufacturing, smart traffic, smart grid, and smart campus
- (2) Analyzes the security and privacy risks faced by 5G applications, including privacy leakage in the eMBB scenario, DDoS attacks in the uRLLC scenario, and remote control in the mMTC scenario
  - (3) Analyzes the existing work for 5G application security, including security standards, security authentication frameworks and protocols, network slicing, and MEC security mechanisms. Particularly, secondary authentications for industry customers and three-factor authentications for mobile lightweight devices are studied
  - (4) Provides the system reference architecture for 5G applications, including the device layer, network layer, platform layer, and service layer, and summarizes security and privacy goals and corresponding solutions layer by layer
  - (5) Summarizes some specific suggestions in typical application scenarios, including secure deployment of edge computing node in the eMBB scenario, preventing application data from tampering/falsification/replay attacks in the uRLLC scenario, and lightweight equipment authentication in the mMTC scenario
  - (6) Provides a use case of industrial terminal access control for 5G application security by triple authentication

The abbreviations in Table 1 are applied in this paper.

## 2. Applications Enabled by 5G-Related Techniques

5G enables a variety of intelligent applications, including smart manufacturing, smart traffic, smart grids, and smart campus. In Figure 3, the blue points are the typical 5G

TABLE 1: Abbreviations.

Abbreviations	Explanation
3GPP	3rd generation partnership project
5G	5th generation mobile network
AF	Application function
AI	Artificial intelligence
AMF	Access and Mobility Management Function
API	Application programming interface
AUSF	Authentication server function
CPE	Customer premise equipment
DDoS	Distributed denial of service
eMBB	Enhanced mobile broadband
EMS	Element management system
IEC	International Electrotechnical Commission
IMSI	International mobile subscriber identity
IoT	Internet of things
IoV	Internet of vehicles
IPS	Intrusion prevention system
ISO	International Organization for Standardization
ITU	International Telecommunication Union
LAN	Local area network
LCS	Location services
LTE	Long term evolution
MANO	Management and orchestration
MEC	Multiaccess edge computing
mMTC	Massive machine type communications
NEF	Network exposure function
NFV	Network function virtualization
NSSAI	Network slice selection assistance information
NSSF	Network slice selection function
PCF	Policy control function
PDCCP	Packet data convergence protocol
RAN	Radio access network
RBAC	Role-based access control
SBA	Service-based architecture
SDN	Software-defined network
SMF	Session Management Function
SUCI	Subscription concealed identifier
UDM	Unified data management
UE	User equipment
UPF	User Plane Function
uRLLC	Ultrareliable and low latency communications
VR/AR	Virtual reality/augmented reality
WAF	Web application firewall
WLAN	Wireless local area network

applications and the grey points are some specific use cases of these applications.

**2.1. 5G Enabled Smart Manufacturing.** Smart manufacturing, today, is the ability to continuously maintain and improve performance, with intensive use of information, in response to the

changing environments [12]. The use cases of 5G technology in the field of intelligent manufacturing are listed below.

**2.1.1. eMBB Scenario.** Using 5G high-bandwidth features and edge computing technology, collecting terminal-side video to the cloud for deep analysis, such as defect detection, OCR decoding, AR assistance, VR complex assembly, production safety behavior analysis, and 5G PLC.

**2.1.2. uRLLC Scenario.** Utilizing 5G low-latency features, network slice, edge computing, and other new technologies to ensure network quality for remote and precise control, such as engineering machinery remote control, AGV control, robot control, and on-site production line equipment control.

**2.1.3. mMTC Scenario.** Using 5G mass-connection, high-bandwidth characteristics, and edge computing technology, collecting sensor data in the factory and transmitting it to the cloud for deep analysis, such as 5G large-scale data collection.

**2.2. 5G Enabled Smart Traffic.** Smart traffic covers vehicles, road infrastructure, traffic management facilities, transportation planning, digital transportation platforms, and various transportation-based applications [13]. The use cases of 5G technology in the transportation industry [14] are listed below.

**2.2.1. eMBB Scenario.** Based on 5G high-bandwidth transmission capabilities, using high-definition video capture and transfer back to the application platform to perform face recognition, such as passenger behavior safety analysis and passengers exit without perception of smart train station.

**2.2.2. mMTC Scenario.** Based on the 5G massive connection characteristics, connect various types of traffic sensors and other IoT devices, to analyze the health status of traffic infrastructure, and timely alert traffic conditions by analyzing various types of data received, such as infrastructure monitoring and inspection, smart subway inspections and maintenance, and warning and management of smart roads.

**2.2.3. uRLLC Scenario.** Based on the high bandwidth, low latency, and massive connection characteristics of 5G, new technologies such as network slicing and edge computing are used to meet the high requirements of unmanned and remotely controlled driving, such as autonomous driving, smart ports, and smart airport.

**2.2.4. Others.** Based on the user's access to the 5G base station, analyze the pedestrian flow within the coverage of the base station, such as smart train station traffic transfer linkage and smart subway passenger flow analysis; based on the 5G base station's precise positioning function, to provide precise positioning services for vehicles and people, such as high-precision positioning and high-precision indoor navigation.

**2.3. 5G Enabled Smart Grid.** Smart grid uses two-way flows of electricity and information to create a widely distributed automated energy delivery network [15]. The use cases of 5G technology in the smart grid industry [16] are listed below.

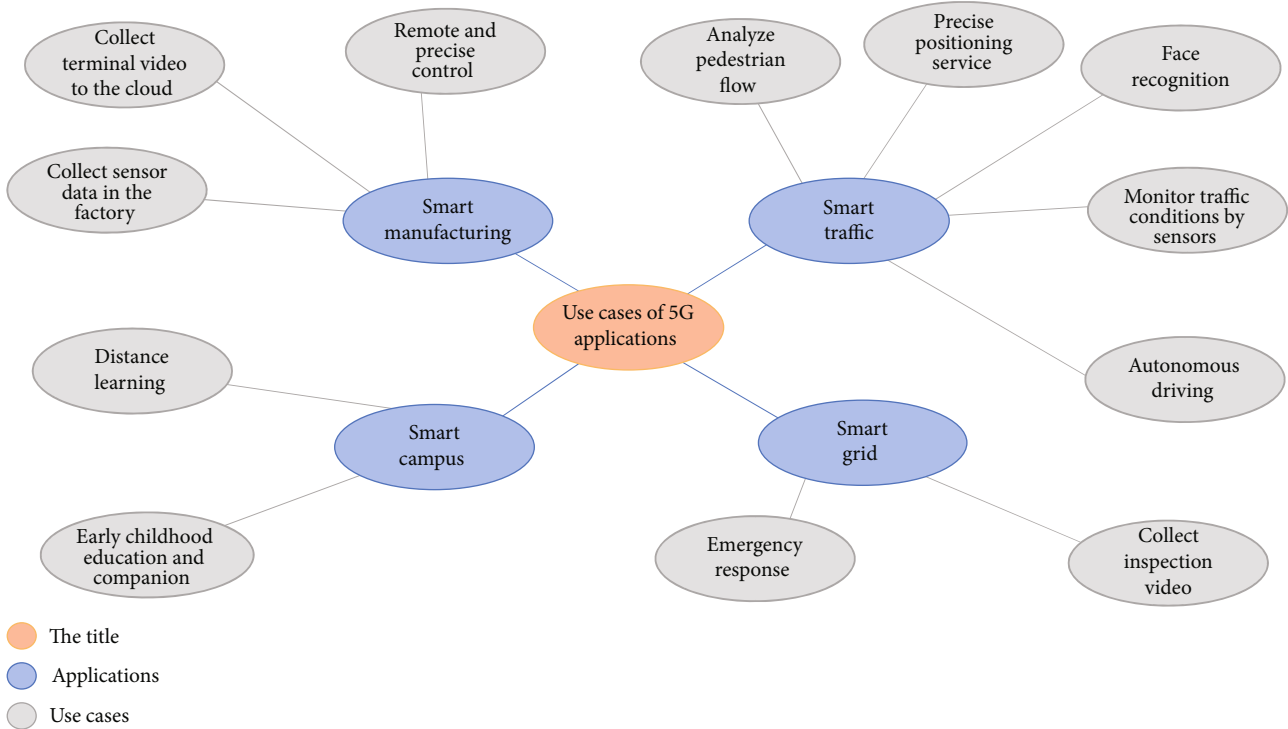


FIGURE 3: Use cases of 5G applications.

**2.3.1. uRLLC Scenario.** Based on 5G low-latency features, slicing, edge computing, and other new technologies, ensure emergency response of the power grid, such as distribution network differential protection, distribution network PMU, and precise load control.

**2.3.2. mMTC Scenario.** Based on 5G mass-connection, high-bandwidth characteristics, and network slicing, edge computing technology, collect inspection video and transmit to the cloud for deep analysis, such as distribution automation of FTU, DTU, and TTU, advanced metering, intelligent inspection, and power grid emergency communications.

**2.4. 5G Enabled Smart Campus.** Smart campus refers to a smart campus based on the Internet of things, which integrates work, study, and life. This integrated environment takes various application service systems as the carrier and fully integrates teaching, scientific research, management, and campus life.

**2.4.1. eMBB Scenario.** Using 5G high-bandwidth features, network slicing, and edge computing technologies for distance learning and AR content dissemination; using 5G slicing technology to carry out applications such as early childhood education, companion robots, and 5G infant growth assessment.

### 3. Risk Analysis of 5G Applications

**3.1. General Risks in 5G Applications.** Security risks for general 5G applications mainly come from the device, network, edge, cloud, and centralized security O&M, as seen in Figure 4.

- (i) Major security risks on the terminal side include unauthorized terminal access, abuse of authorized SIM cards, and attacks and control of authorized terminals
- (ii) Major security risks on the network side include network slicing isolation, misuse of slice resources, and theft and tampering of user-plane information
- (iii) Security risks on the edge MEC side include vulnerabilities on the MEC platform, untrusted applications on the MEC, and attacks on the MEC from the Internet, enterprise cloud, and OM plane
- (iv) Security risks on the enterprise private cloud include MEC-based attacks on the enterprise intranet and enterprise communication theft or tampering
- (v) Finally, from the perspective of O&M management, there are risks such as security posture awareness failure, unified management of security devices and policies, and lack of O&M audit

#### 3.2. 5G Specific Risks in Typical Usage Scenarios

**3.2.1. eMBB Scenario.** eMBB focuses on applications with extremely high bandwidth requirements. Currently, 4K/8K high-definition video and mobile roaming immersive services based on virtual reality (VR) and augmented reality (AR) have become the main application forms of eMBB, which mainly includes the following security risks:

- (i) *Failure of Monitoring Means.* eMBB applications produce huge volumes of traffic which would make it



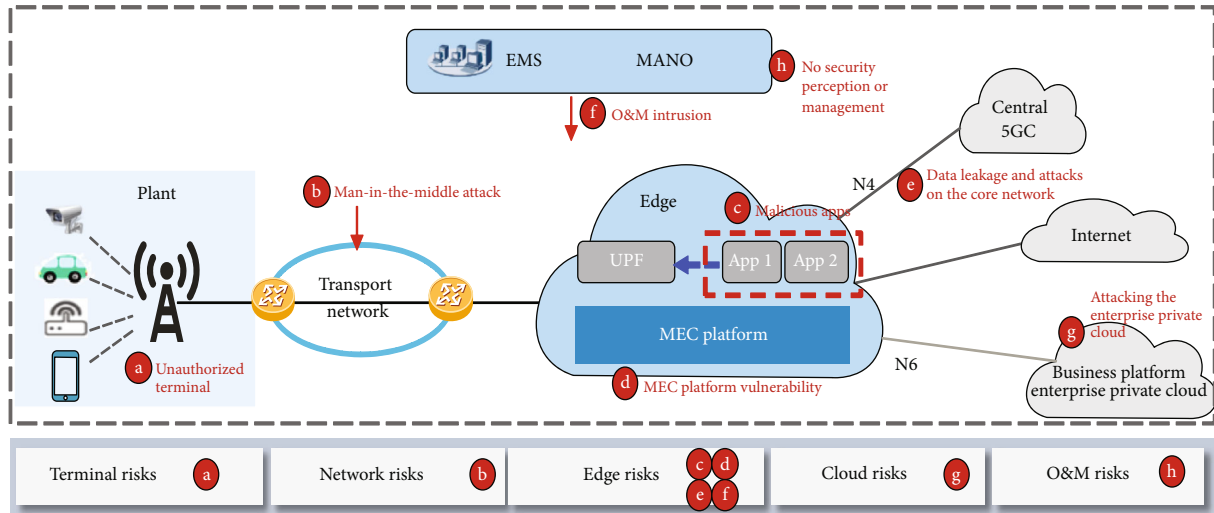


FIGURE 4: Risks to 5G applications in an end-to-end view.

extremely difficult for security devices such as firewalls and intrusion detection systems deployed in existing networks to ensure adequate security protection when it comes to traffic detection, radio coverage, and data storage [17].

- (ii) *User Privacy Leakage.* eMBB services (such as VR/AR) contain a large amount of user privacy information, such as personal information or identification, device identification, and address information, and the openness of 5G networks has increased the probability of leakage of private information [18].

**3.2.2. uRLLC Scenario.** uRLLC focuses on services that are extremely sensitive to latency, such as autonomous driving/assisted driving, remote control, and industrial Internet. Low latency and high reliability are the basic requirements. For example, if the internet of vehicles is subject to security threats in communications, it may cause danger of life. Therefore, uRLLC services require high-level security without additional communication delays. The main security risks are as follows:

- (i) *DDoS Attacks.* Attackers may use DoS/DDoS attacks to cause network congestion or communication interruptions, causing failure of services
- (ii) *Data Security Risks.* Attackers use vulnerabilities in devices and protocols along network data transmission paths (5G air interfaces, core networks, and the Internet) to tamper with/forged/replay application data [14], causing the drop of data transmission reliability and harm to normal application operations

**3.2.3. mMTC Scenario.** The 5G mMTC scenario supports IoT applications with massive devices being connected, such as smart transportation, smart grids, and smart cities. Due to the low cost, mass deployment, and limited resources (such as processing, storage, and energy) of the Internet of things [19], the following security risks are common to IoT devices:

- (i) *Counterfeit Terminals.* The IoT terminal has limited resources and weak processing and computing capabilities. Therefore, it is likely that authentication would not be performed or a simple method has to be adopted [20, 21], which brings opportunities for counterfeit terminals, causing confusion for the operation of IoT applications
- (ii) *Data Tampering.* Attackers may tamper with application data by exploiting weaknesses of the terminal and cloud/edge platform
- (iii) *Data Eavesdropping.* The data collected by IoT terminals deployed in special environments (such as home environments and medical environments) involves user privacy. Weaknesses along data transmission paths may lead to user privacy breaches
- (iv) *Remote Controls.* Attackers may remotely access and control IoT terminals through software and hardware interfaces by taking advantage of the simplicity of IoT terminals and weak security protection capabilities, and then use the captured terminals to launch network attacks [22–26].

Based on the above analyses, typical security and privacy risks of use cases in 5G vertical applications are listed in Table 2.

## 4. Related Work on Security of 5G Applications

**4.1. Security Standards on 5G Applications.** For 5G applications, the R16 standard released by 3GPP further enhances the quality and efficiency of 5G applications. For example, for Industrial Internet, new technologies are introduced to support 1 ms synchronization accuracy and 0.5-1 ms air interface delay, which can achieve end-to-end lower latency and higher reliability. For internet of vehicles, it supports the direct connection communication of V2V (vehicle-to-vehicle) and V2I (vehicle-to-infrastructure). By a variety of communication

TABLE 2: The security and privacy risks of typical applications.

Typical applications	Specific use cases	Risks examples
Smart manufacturing	AR assistance, VR complex assembly	Counterfeit terminals and failure of monitoring means
	Collecting sensor data of IoT device	Data tampering and data eavesdropping
	Remote control of engineering equipment	DDoS attacks and remote control
Smart traffic	Connected vehicles	DDoS attacks and data security risks
	Passenger behavior safety analysis	Failure of monitoring means
Smart grids	Distribution network differential protection and precise load control	DDoS attacks
	Customized network slice to satisfy the low time latency requirement	Counterfeit terminals and management of network slices
Smart campus	Distance learning and AR content dissemination	Failure of monitoring means and user privacy leakage
	Front-projected holographic display	Failure of monitoring means and user privacy leakage

methods such as multicast and broadcast, as well as technologies such as optimized perception, scheduling and retransmission realize V2X (vehicle-to-everything) to support vehicle networking, semiautomatic driving, epitaxial sensors, remote driving, and other IoV (internet of vehicles) scenarios. For industry applications, the introduction of a variety of 5G air interface positioning technologies improves positioning accuracy by more than ten times and reach meter level.

5G applications involve various roles such as communication network providers, industry application providers, and security regulatory agencies. Currently, standards are mainly developed through collaboration between relevant parties to ensure application security. For 5G application security, major international standards organizations and industry associations have carried out research work, as shown in Table 3 [27–29].

*4.2. Authentications in 5G Applications.* Security authentications face higher requirements in 5G applications. On the one hand, in order to protect the application data of power, industry, finance, and other important fields carried by 5G network, the concept of secondary authentication is proposed, that is, the authentication to establish data channel for accessing specific business after user authentication for access network. On the other hand, with the rapid development of 5G applications, mobile lightweight devices including laptops, smartphones, smartwatch, and other wearable devices are increasingly popular. It is necessary to concern the authentication for mobile lightweight devices and guarantee user privacy.

*4.2.1. Secondary Authentications for Industry Customers.* In the implementation scheme based on the 3GPP standard [28], the protocol stack between the user terminal and the AAA (authentication, authorization, and audit) server is shown in Figure 5. The secondary authentication protocol between the UE and the AAA server is carried by EAP (Extensible Authentication Protocol). During the interaction of the secondary authentication protocol, AN (access network), AMF (Access and Mobility Management Function), SMF (Session Management Function), UPF (User Plane

Function), and other network elements will not parse the secondary authentication protocol and can realize end-to-end secondary certification of users in enterprise and industry.

Generally, industry customers deploying 5G applications can directly complete the secondary authentication by algorithms and protocols provided by telecommunication operators. 3GPP [28] defines a series of standard secondary authentication protocols, including PAP (Password Authentication Protocol), CHAP (Challenge Handshake Authentication Protocol), PPP (Point-to-Point Protocol), AKA (Authentication and Key Agreement), and TLS (Transport Layer Security). PAP and CHAP use a relatively simple authentication mechanism. AKA and TLS are based on cryptographic algorithms and have designed a relatively blameless protocol to achieve user access authentication. In addition, based on the openness of 5G network capabilities, the AKMA [29] mechanism was proposed. The mechanism can provide authentication and session key negotiation services for third-party applications based on the access authentication system of the USIM card and carrier network and establish secure transmission channels from terminals to applications.

Users with high-security requirements can also take advantage of the openness of 5G network capabilities and the industry-oriented feature and use customized secondary authentication algorithms and protocols to realize the self-controllable secondary identity authentication of the enterprise or industry. Chen et al. [30] proposed a customized secondary authentication protocol, mainly using mobile terminals to collect biometric information such as fingerprints and irises of users and combined with the challenge-response identity authentication mechanism for identity authentication. Li et al. [31] proposed a secondary authentication protocol based on a symmetric cryptosystem that improves existing protocols such as AKA and provides user identity information protection, message integrity protection, and two-way authentication. Liu et al. [32] proposed an online identification technique with biological characteristic authentication and multimedia signal fast encoding over 5G to deal with the explosive growth in mobile data generated by huge equipment connections and a large number of new business and application scenarios.

TABLE 3: Security standards on 5G applications.

Organization	Technical standards and reports
3GPP	3GPP TS 22.261 service requirements for the 5G system: (i) R15 focuses on supporting eMBB services and basic uRLLC services (ii) R16 enhances the ability and efficiency of network to support eMBB (iii) R16 focuses on improving support for vertical industry applications, especially uRLLC and mMTC services.
	3GPP TS 33.501 security architecture and procedures for 5G system: (i) The application layer access authentication and secure channel establishment in the IoT (ii) The solution of authentication and session key management for upper-layer applications provided by 5G security certificate.
	3GPP TR 33.819 study on security enhancements of 5GS for vertical and local area network (LAN) services: (i) The security requirements and solutions of the 5G vertical industry.
	3GPP TR 33.814 study on the security of the enhancement to the 5GC (5G core network) location services (LCS): (i) The security threats and requirements and solutions of 5GC LCS.
	3GPP TR 33.836 study on security aspects of 3GPP support for advanced V2X services: (i) The security threats and requirements and solutions of IOV.
ITU	3GPP TR 33.825 study on the security of ultrareliable low-latency communication (URLLC) for 5GS (i) The security requirements and solutions of the uRLLC scenarios.
	ITU-T X.1373 secure software update capability for intelligent transportation system communication devices: (i) The software security update between the remote update server and the vehicle couplet (ii) The process and content recommendations for security update.
ISO	Criteria for the assessment of information security of connected vehicles based on ISO/IEC 15408: (i) The security threats and security goals faced by connected vehicles (ii) The security requirements and security function components.

*4.2.2. Three-Factor Authentications for Mobile Lightweight Devices.* Mobile lightweight devices can conveniently access cloud servers for online payment, video chatting, e-commerce, etc. At the same time, the openness of wireless network communication will also bring risks to the security and privacy of user data, so authentication for mobile light devices should be considered. Authentication and Key Agreement (AKA) protocols based on public key technology provide a secure communication mechanism for 5G application environments. It is essential to establish an AKA protocol to protect the conversation between mobile lightweight devices and remote servers. In 2018, Wang et al. [33] described the identity-based AKA protocols for privacy preserving of mobile devices and pointed out corresponding challenges. Moreover, Xiao et al. [34] proposed an improved AKA protocol based on chaotic maps and then a series of AKA protocols based on chaotic maps [35–37] have been proposed.

In addition, it is generally believed that the three-factor AKA protocol has better security performance than single-factor and two-factor protocols. Since the existing three-factor AKA protocol cannot meet all the security requirements, it has become a research focus in recent years. Biometrics including fingerprint, face, iris, and others are invariable physiological characteristics that people own, and nowadays more and more mobile lightweight devices have the function of biometric recognition. In the face of stringent security requirements, the combination of traditional AKA protocol and the third authentication factor (i.e., biometrics) can achieve higher security [38, 39]. In order to solve the common security problems in the existing three-factor AKA protocol, Qiu et al. [40] designed a new three-factor AKA protocol by combining biometrics with

chaotic mapping, using “Fuzzy Verifiers” and “Honeywords,” which can achieve semantic security and meet the security evaluation criteria. Finally, it is proved that the new three-factor AKA protocol is more practical on mobile lightweight devices.

*4.3. Other Research Focuses.* As for the security architecture of 5G application, GTI (Global TD-LTE Initiative) released the security reference architecture of 5G smart city [41]. Zhou et al. [42] proposed the service architecture, PKI architecture, and multi-PKI mutual trust mechanism for 5G V2X communication security. Wang and Liu [43] analyzed 5G applications for special industries with high security levels and the security enhancement requirements and proposed a design scheme of security architecture based on special industry slices.

*4.3.1. MEC.* As for key security technologies of 5G application, MEC is the technology most closely related to 5G applications. According to ETSI [44], MEC architecture is divided into system level and host level. There is a remarkable resemblance of risks between MEC and cloud infrastructure, so their security measures are also similar. He et al. [45] proposed to enhance the isolation and access control by standardizing the configuration of infrastructure and application system, so as to improve the security protection ability of MEC nodes. At the same time, strengthen the security control of MEC applications. Zhuang et al. [46] analyzed the security threats, protection framework, and scheme of MEC from aspects of infrastructure, MEC platform, ME app, MEC scheduling and management system, and gateway of data plane.

*4.3.2. Network Slicing.* Network slicing is another important technology of 5G. Zhou [47] proposed four network slicing

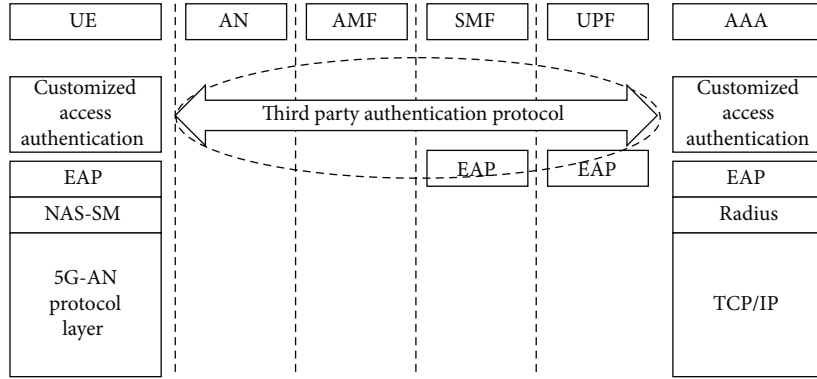


FIGURE 5: End-to-end protocol stack for secondary authentication [28].

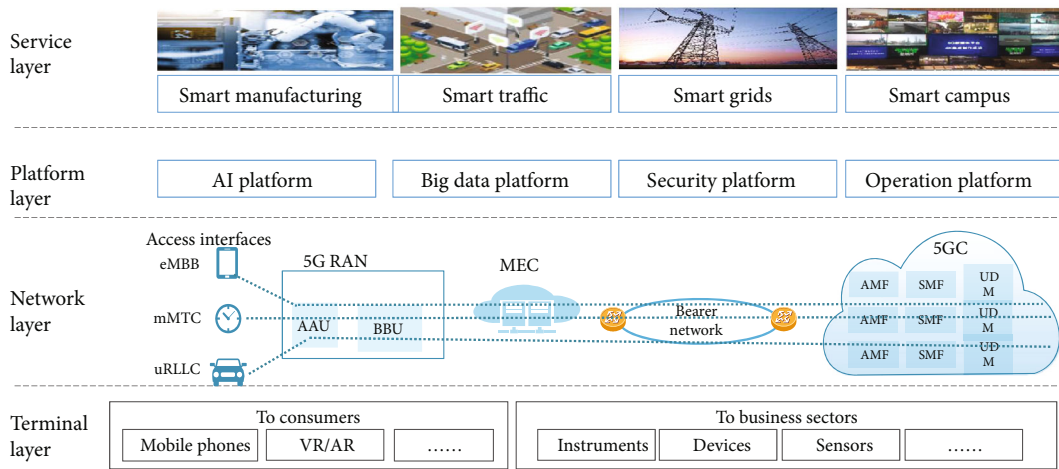


FIGURE 6: Reference architecture of 5G applications.

deployment schemes according to different requirements of cost, QoS, security levels, and network topology flexibility. Liu et al. [48] elaborated the existing risks of network slicing from the framework, management model, and implementation technology of network slicing and provided differentiated security services for 5G network slicing by establishing a security model. Chen et al. [49] proposed technical solutions to the security threats caused by the introduction of 5G into network slicing and proposed the security isolation of network slices, the secure access of terminal access slices, the security construction of network slices, and the security communication within the slices. The thesis [50] proposes 5G-SSAAC (5G Slice-Specific AAC), which enables 5G networks to provide various AAC mechanisms to the 3rd parties according to their security requirements.

### 5. Security and Privacy Solutions in a Systematic View

5G applications can be modelled into the terminal layer, network layer, platform layer, and service layer [51], as shown in Figure 6.

Each layer has corresponding security goals and solutions, as shown in Table 4.

**5.1. Solutions on Terminal Layer.** A large number of 5G terminals have low power consumption, as well as limited computing and storage resources, which makes the deployment of complex security policies and control over the software difficult. Consequently, these limitations make the terminals become easy and likely targets to be hacked [24].

**5.1.1. Prevent and Defend against DDoS Attacks.** DDoS attacks may be initiated by hacked terminals or unintentionally caused by software defects or network faults. It is recommended that security defense mechanisms to be built at the network level for attack detection and self-protection to ensure that any DDoS attacks can be detected in time. Besides, active preventive measures are recommended in terminal exception handling and signaling registration.

**5.1.2. Prevent Various Damage Caused by Exploited Terminals.** For the prevention of risks brought by terminal hacking, it is recommended that certain security capabilities such as SSH security login, TLS transmission encryption, and built-in security chip are being built in terminals in terms of access

TABLE 4: Security and privacy solutions for 5G applications.

Layer	Targets	Security and privacy solutions
Terminal layer	Prevent and defend against DDOS attacks	(i) Attack detection and self-protection mechanisms (ii) proactive preventive measures
	Prevent various damage caused by exploited terminals	(i) Access authentication [52, 53] on the operation and maintenance side (ii) encryption protection on the signaling/data plane
Network layer	Base station air interface security	(i) Defense eavesdropping and tampering of user data (ii) defense DDOS attack from air interface (iii) pseudo base station detection [54]
	MEC security	(i) Physical environment security control (ii) enterprise and operator network isolation
	5GC security	(i) Manage operation and maintenance plane security (ii) network north-south border security (iii) east-west security within the network (iv) cloud-based security of the core network
	Bearer network security	(i) Network redundant design (ii) account authority management and access authentication (iii) increase security measures on control protocols (iv) user plane security encryption
	5G slice security [55]	(i) Isolation between slices (ii) secure access and use of slices (iii) privacy protection
Platform layer	The security of communications interfaces.	(i) Routine maintenance of various account passwords (ii) encryption of communication interfaces
	The security of platform data.	(i) Data availability, integrity, and privacy
Service layer	Software security of the application	(i) Vulnerability scanning of the software (ii) software operation logging (iii) highly available disaster recovery of software systems
	O&M security of the application	(i) Security constraints and controls for application system (ii) physical security control (personal access control) of O&M operations office/machine room, etc.

authentication [25, 26] on the management and O&M plane as well as encryption protection on the signaling/data plane.

*5.2. Solutions on Network Layer.* From the perspective of network components, the noteworthy aspects of network layer security include security in the RAN base station air interfaces [56], MECs, 5G Core, bearer networks, and 5G slices.

*5.2.1. Base Station Air Interface Security.* To prevent user data eavesdropping and tampering, SUCI and air-interface PDCP data packets encryption can be enabled. Besides, a DDOS detection and defense system and a unified rogue base station detection system can be deployed to avoid malicious attacks and interference.

*5.2.2. MEC Security.* To avoid physical attacks and cross-network penetration and infection of network, 5G networks need to focus not only on the physical security control of MEC but also on the isolation between enterprise networks and operator networks. Security facilities such as firewalls and IPS are recommended for network boundary protection [57–63].

*5.2.3. 5GC Security.* For MANO, EMS, etc., an access security control system is suggested to avoid unauthorized management and O&M access. To prevent viruses and OS vulnera-

bilities caused by O&M terminals, desktop cloud terminals can be used. For the north-south border security of the network, firewalls, sandboxes, WAF, IPS, and anti-DDoS devices can be deployed in the data center. For the east-west security, network microsegmentation, whitelist ACL, and network traffic probe ought to be deployed. Finally, it is recommended that host security scanning and hardening are being routinely implemented, and monitoring software is being deployed at the hypervisor level of servers to prevent VM escape [64–67].

*5.2.4. Bearer Network Security.* For network planning and design, redundancy design needs to be adopted to avoid single points of failure. Permission management and access authentication of accounts and passwords need to be implemented. Security measures such as MD5 authentication or SSL encryption can be configured to avoid possible routing protocol attacks such as BGP routing hijack attacks. Besides, IPsec encryption can be deployed to ensure the integrity of network data packets, to prevent illegal traffic interception or network replay attacks.

*5.2.5. 5G Slice Security.* The security of 5G network slicing [55] needs to be protected by isolation between slices. Besides,

TABLE 5: Countermeasures against security and privacy risks in 5G applications.

Risks	Countermeasures	Related layer
eMBB scenario		
Failure of effective monitoring means	(i) Application traffic monitoring at edge computing [63] nodes, suspension of high-risk services in specific cases	(i) Network layer
User privacy leakage risk	(i) Perform secondary identity authentication and authorization between the terminal and the eMBB application service platform (ii) negotiate and manage the service layer key to encrypt and protect user data (iii) physical isolation or encryption (iv) network slicing [55] or data dedicated line	(i) Terminal layer (ii) network layer (iii) service layer
uRLLC scenario		
DDoS attack risk	(i) Two-way identity authentication between the user terminal and the application servers (ii) deploy anti-DDoS capabilities	(i) Network layer (ii) terminal layer
Data security risk	(i) Security capabilities deployed at edge computing [51], as well as data integrity protection, timestamp, serial number, etc. [18];	(i) Network layer
mMTC scenario		
Counterfeit terminal	(i) Using lightweight security algorithms [52, 53, 73], simple and efficient security protocols to implement two-way authentication	(i) Terminal layer
Data tampering and eavesdropping	(i) Encrypt and protect the integrity of sensitive application data generated by IoT terminals [18]	(i) Terminal layer
Remote control	(i) Deploy security monitoring methods [68, 69] to timely detect and prevent massive IoT devices from being controlled	(i) Terminal layer

secure access and use of slices are also recommended. Access to a corresponding 5G network slice requires dual authentications and authorizations by the slice user (such as a government agency or an industrial mining enterprise) and the operator, ensuring legal access and use of slice resources. Moreover, the privacy protection of Network Slice Selection Assistance Information (NSSAI) needs to be provided.

**5.3. Solutions on Platform Layer.** The platform layer covers various intelligent analysis and processing AI platforms, big data platforms, and IT middle ground [68, 69]. The security of this layer includes the following aspects.

**5.3.1. The Security of Communications Interfaces.** In general, communication interface security at the platform layer mainly focuses on the routine maintenance and management of various accounts and passwords, such as regular password changes and password complexity requirements and the encryption of communications interfaces such as TLS.

**5.3.2. The Security of Platform Data.** The security of data at the platform layer involves the security of various basic data collected and stored by the big data platform, including data availability, integrity, and privacy. Availability is guaranteed by technologies such as data redundancy. Integrity is guaranteed by technologies such as data verification. For privacy, as the data amount is usually huge, more effective access control and security audit are required.

**5.4. Solutions on Service Layer.** The security of the service layer consists of various application system software security and secure O&M of application systems.

**5.4.1. Software Security of the Application.** Application system software security mainly involves scans for vulnerabilities and the improvement of software security (including the application software itself, OS databases, and other software systems), software operation logging, and software system high availability (HA) disaster recovery deployment (such as dual-host backup).

**5.4.2. O&M Security of the Application.** Secure O&M of application systems focus more on the operation and use of application systems and the security constraints and control of information on the operation management personnel, for example, application system login accounts and passwords, multifactor authentication for important and sensitive operations, permission-based operation access control, and physical security control of personnel access of O&M operations offices and equipment rooms.

## 6. Countermeasures against Security and Privacy Risks in 5G Applications

Based on the systematic security and privacy solutions proposed above, the following specific security measures are recommended for 5G application service developers and providers in different application scenarios [70–72]. The related layers in the reference architecture to deploy these countermeasures are also suggested (see Table 5).

**6.1. eMBB Scenario.** Security risks in the eMBB scenario mainly include failure of effective monitoring means and user privacy leakage, and the countermeasures are as follows:

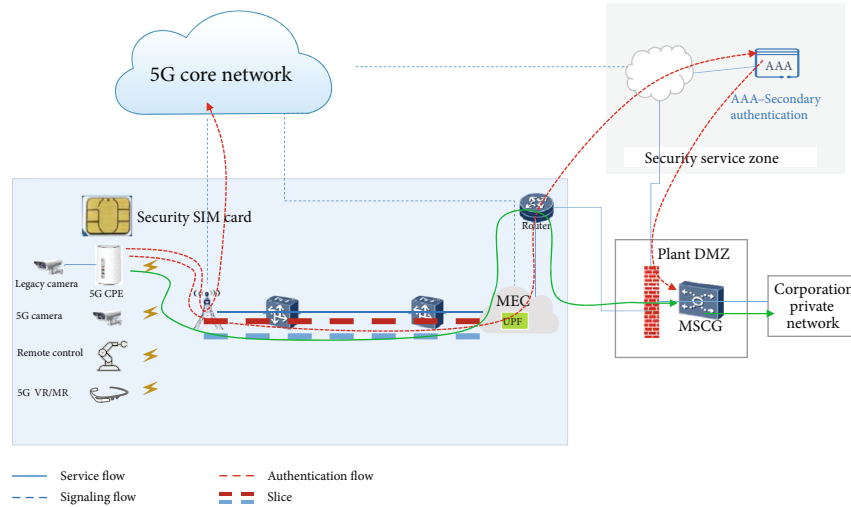


FIGURE 7: A use case of industrial terminal access control.

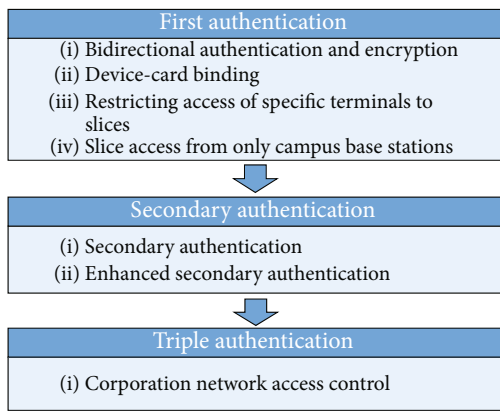


FIGURE 8: Terminal access control solutions by triple authentication.

- (i) Deploy application traffic monitoring at edge computing [63] nodes and support the suspension of high-risk services in specific cases
- (ii) The secondary authentication and key management mechanism are used to perform secondary identity authentication and authorization between the terminal and the eMBB application service platform to ensure the authenticity of the terminal and platform identity and the legality of the application. At the same time, negotiate and manage the service layer key between the two sides to encrypt and protect user data, thus preventing attackers from eavesdropping
- (iii) In applications with high-security requirements, the user plane of the 5G network can be protected by physical isolation or encryption to ensure the security of user data transmission between network functions
- (iv) The network slicing or data dedicated line is used between the operator's 5G core network and the eMBB application service platform to establish a secure data transmission channel to ensure the security of user business data transmission

6.2. *uRLLC Scenario.* Security risks in the uRLLC scenario mainly include the DDoS attack and the data security risk, and the corresponding countermeasures are as follows:

- (i) Establish a two-way identity authentication mechanism between the user terminal and the application server to prevent fake users from establishing connections
- (ii) Deploy anti-DDoS capabilities to prevent network congestion, wireless interference, and communication link disruptions
- (iii) Through the security capabilities deployed at edge computing, as well as data integrity protection, timestamp, serial number, and other mechanisms, to prevent application data from being tampered/falsified/replayed and ensure the reliability of data transmission [60]

6.3. *mMTC Scenario.* Security risks in the mMTC scenario mainly include the counterfeit terminal, data tampering and eavesdropping, and remote control, and the corresponding countermeasures are as follows:

- (i) Using lightweight security algorithms, simple and efficient security protocols to implement two-way authentication between IoT terminals and the network to ensure that the access terminals are secure and reliable
- (ii) Encrypt and protect the integrity of sensitive application data generated by IoT terminals to prevent attackers from eavesdropping, tampering, forging, and replaying business data on the transmission path
- (iii) Deploy security monitoring methods [68, 69] to timely detect and prevent massive IoT devices from being controlled, to prevent these devices from being used maliciously, such as launching DDoS attacks on air interfaces and service platforms, causing network congestion and causing mMTC services to fail

TABLE 6: Conclusions of the paper.

No.	Contributions	Results
1	Summarizes the technical characteristics and typical application scenarios of 5G	(a) The features summarization introduced by new technologies (b) vertical applications introduction: (i) smart manufacturing (ii) smart traffic (iii) smart grids (iv) smart campus
2	Analyzes the security and privacy risks faced by 5G applications	(a) General security risks analysis in 5G applications (b) 5G specific risks analysis in typical usage scenarios: (i) privacy leakage in the eMBB scenario (ii) DDoS attacks in the uRLLC scenario. (iii) remote control in the mMTC scenario
3	Analyzes the existing work for 5G application security	(a) Security standards introduction on 5G applications (b) authentications analysis in 5G applications: (i) secondary authentications for industry customers (ii) three-factor authentications for mobile lightweight devices (c) other research focuses analysis: (i) MEC (ii) network slicing
4	Analyzes the reference architecture and summarizes security solutions for 5G applications	(a) The system reference architecture analysis (b) security and privacy solutions: (i) solutions on terminal layer (ii) solutions on network layer (iii) solutions on platform layer (iv) solutions on service layer
5	Summarizes security measures in typical scenarios and proposes specific suggestions	(a) Specific suggestions for security and privacy: (i) secure deployment of edge node in eMBB scenario (ii) preventing data from various attacks in uRLLC scenario (iii) lightweight equipment authentication in mMTC scenario
6	Provides a use case of industrial terminal access control	(a) The description of the use case and its security requirements (b) terminal access control solutions

## 7. A Use Case of Industrial Terminal Access Control

*7.1. Introduction and Security Requirements.* This is a case of industrial terminal access control, as shown in Figure 7. The services include industrial machine vision for quality inspection that requires high bandwidth, automatic robot control, crane remote control, and unmanned transportation with real-time control requirement. Considering that the campus coverage area does not need to be large and high security is required when data cannot be transmitted out of the campus, the UPF and MEC are deployed at the local edge, and different service networks are isolated.

This case involves several security requirements on terminal access controls.

- (i) Prevent terminals such as 5G CPE, AGV, and gantry crane being attacked or illegally controlled
- (ii) Prevent CPEs being accessed by fake terminals, so that legal terminals (such as PLC) and the central control system would not be attacked
- (iii) Prevent the SIM card from being removed from the legal terminal and inserted into a malicious terminal

*7.2. Terminal Access Control Solutions.* With the purpose that only authorized terminals can access the enterprise private network, the carrier and enterprise jointly provide triple authentication, as shown in Figure 8.

First, carriers enable 5G AKA-based bidirectional authentications on the RAN side, leading the bidirectional authentication and encryption (5G AKA standard) between the 5G CPE/5G camera and the 5G network to prevent the fake terminals from accessing. Legacy cameras also must pass AAA authentication before accessing the CPE. Besides, configure the terminal whitelist and device-card binding on the core network to prevent unauthorized terminals and legal SIM card abusing. 5G CPE configured with MAC address list that allows access of traditional cameras. Then, the core network binds the network slice to the terminal identity and the physical location that the terminal can access and also restricts access of specific terminals to slices. The mapping between IMSI and slice S-NSSAI is configured on the 5GC. Only terminals in the campus IMSI list can access slices. Mapping between the TAI (Tracking Area Identifier) list and campus slice S-NSSAI configured on the 5GC, and only authorized terminals can access the enterprise private network within the campus.

Second, enterprises deploy the AAA system in the security service zone to provide secondary authentication for



terminals accessing the slice in Username-Password mode. By using AAA system and security SIM card technology, terminals and applications that have high-security requirements can improve secondary authentication strength. Here, the security SIM card is a USIM-based card with a built-in USB key function. It is based on the PKI digital certificate system. The key is stored in the security chip of the SIM card and cannot be copied, repudiated, or tampered with.

Third, the enterprise can deploy the multiservice access gateway (MSCG) at the intranet border. The MSCG grants the access rights of terminals to the enterprise private network only after the terminals pass the second authentication.

With the implementation of the above schemes, the factory campus has denied 10412 access queries from untrusted terminals during the past 6 months.

## 8. Conclusions

5G is deeply integrated with social life and vertical industries, and the security and privacy of the 5G ecosystem are largely influenced by application developers and service providers, as well as network operators and equipment suppliers. The achievement of security and privacy in 5G applications requires a comprehensive and systematic design, as well as the deployment of proper security measures according to the specific application scenarios and the needs of the industry.

This paper makes contributions in the research of security and privacy in 5G-enabled applications, as shown in Table 6. In view of numerous 5G applications, such as smart manufacturing, smart transportation, smart grid, and smart campus, this paper analyzes general security risks from devices, networks, edges, and other aspects, as well as specific risks in typical usage scenarios. As a result, readers will have a more comprehensive grasp of security risks in 5G applications. Besides, the existing related work for 5G application security is analyzed, including security standards, authentications, network slicing, and MEC. In particular, secondary authentications for industry customers and three-factor authentications for mobile lightweight devices are researched. After that, the reference architecture of 5G applications is analyzed, and security solutions are summarized in a systematic view. In addition, we also analyze the security and privacy risks for 5G applications in eMBB, uRLLC, and mMTC scenarios and summarize corresponding countermeasures. Finally, a use case of industrial terminal access control is studied, which enhances readers' understanding of specific 5G application security risks and solutions. On the whole, this paper conducts a comprehensive study on security and privacy in 5G applications, which strengthens readers' risk awareness and security capabilities and generates a positive impact on the healthy and sustainable development of various applications in 5G era.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

This paper is supported by the construction project of the Joint Laboratory for Mobile Learning, Ministry of Education-China Mobile Communications Corporation (no. ML2012934).

## References

- [1] S. E. Elayoubi, J. S. Bedo, M. Filippou et al., "5G innovations for new business opportunities," in *Mobile World Congress, 5G Infrastructure association, Mobile World Congress, Barcelona, Spain, 2017*.
- [2] TS 22261, *Technical Specification Group Services and System Aspects; Service Requirements for the 5G system; Stage 13GPP*.
- [3] 5G Network Architecture and Security, *DCMS Phase 1 5G Testbeds & Trials Programme*, Department for Digital, Culture, Media & Sport. UK, 2018.
- [4] TS 23501, *System Architecture for the 5G System*, 3GPP, 2018.
- [5] GS MEC-002, *MEC Technical Requirements*, ETSI, 2016.
- [6] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, "A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges," *Wireless Networks*, vol. 21, no. 8, pp. 2657–2676, 2015.
- [7] L. Gavrilovska, V. Rakovic, and V. Atanasovski, "Visions towards 5G: technical requirements and potential enablers," *Wireless Personal Communications*, vol. 87, no. 3, pp. 731–757, 2016.
- [8] "Setting the scene for 5G: opportunities & challenges," [https://www.itu.int/en/ITU-D/Documents/ITU\\_5G\\_REPORT-2018.pdf](https://www.itu.int/en/ITU-D/Documents/ITU_5G_REPORT-2018.pdf).
- [9] IMT Vision, *Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, ITU-R M.2083-0, 2015.
- [10] European Commission, *Commission Recommendation of 26.3.2019 Cybersecurity of 5G networks*, European Commission, 2019.
- [11] I. H. S. Markit, *The 5G Economy: How 5G Technology Contribute to the Global Economy*, HIS Technology, 2019.
- [12] K. Jung, B. Kulvatunyou, S. Choi, and M. P. Brundage, "An overview of a smart manufacturing system readiness assessment," in *IFIP Advances in Information and Communication Technology*, pp. 705–712, Springer, Cham, Switzerland, 2016.
- [13] Z. Bao, "Discussing 5G network technologies in smart traffic construction," *China ITS Journal*, vol. 226, no. 1, pp. 81–82 +102, 2019.
- [14] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 772–782, 2017.
- [15] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid — the new and improved power grid: a survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2012.
- [16] N. Saxena, A. Roy, and H. S. Kim, "Efficient 5G small cell planning with eMBMS for optimal demand response in smart grids," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1471–1481, 2017.

- [17] CAICT, IMT 2020(5G), Promotion Group, *5G Security Report*, The China Academy of Information and Communications Technology (CAICT) and IMT 2020(5G) Promotion Group, 2020.
- [18] M. A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, "Security for 4G and 5G cellular networks: a survey of existing authentication and privacy-preserving schemes," *Journal of Network and Computer Applications*, vol. 101, pp. 55–82, 2018.
- [19] K. Fan, Y. Gong, C. Liang, H. Li, and Y. Yang, "Lightweight and ultralightweight RFID mutual authentication protocol with cache in the reader for IoT in 5G," *Security and Communication Networks*, vol. 9, no. 16, 3104 pages, 2016.
- [20] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2018.
- [21] D. He, D. Wang, and S. Wu, "Cryptanalysis and improvement of a password-based remote user authentication scheme without smart cards," *Information Technology and Control*, vol. 42, no. 4, pp. 170–177, 2013.
- [22] GTI, *5G Network Security Consideration White Paper v1.0*, GTI, 2019.
- [23] ENISA, *Threat Landscape for 5G Networks*, European Union Agency for Network and Information Security (ENISA), 2019.
- [24] X. Ji, K. Huang, L. Jin et al., "Review of 5G security technology," *Mobile Communications*, vol. 43, no. 1, pp. 40–45+51, 2019.
- [25] D. Wang, C.-G. Ma, Q.-M. Zhang, and S. Zhao, "Secure password-based remote user authentication scheme against smart card security breach," *Journal of Networks*, vol. 8, no. 1, pp. 148–155, 2013.
- [26] D. Wang, X. Zhang, Z. Zhang, and P. Wang, "Understanding security failures of multi-factor authentication schemes for multi-server environments," *Computers & Security*, vol. 88, p. 101619, 2020.
- [27] 3GPP TS 22 261, *Service Requirements for the 5G System*, The 3rd Generation Partnership Project (3GPP), 2018.
- [28] 3GPP TS 33 501, *Security Architecture and Procedures for 5G System R15 TS 33.501*, The 3rd Generation Partnership Project (3GPP), 2018.
- [29] 3GPP TR 33.835, *Study on Authentication and Key Management for Applications Based on 3GPP Credential in 5G*, The 3rd Generation Partnership Project (3GPP), 2018.
- [30] F. L. Chen, J. Wang, and X. Du, "Primary exploration of 5G secondary identity authentication scheme for enterprise/industry users," *Communications Technology*, vol. 52, no. 7, pp. 1740–1743, 2019.
- [31] C. L. Li, Y. Gu, and J. Wang, "Analysis and design of 5G secondary authentication protocol," *Communications Technology*, vol. 52, no. 7, pp. 1733–1739, 2019.
- [32] X. Liu, P. Wang, Z. Lan, and B. Shao, "Biological characteristic online identification technique over 5G network," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 84–90, 2015.
- [33] D. Wang, H. Cheng, D. He, and P. Wang, "On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices," *IEEE Systems Journal*, vol. 12, no. 1, pp. 916–925, 2018.
- [34] D. Xiao, X. Liao, and S. Deng, "A novel key agreement protocol based on chaotic maps," *Information Sciences*, vol. 177, no. 4, pp. 1136–1142, 2007.
- [35] L. Han, Q. Xie, W. Liu, and S. Wang, "A new efficient chaotic maps based three factor user authentication and key agreement scheme," *Wireless Personal Communications*, vol. 95, no. 3, pp. 3391–3406, 2017.
- [36] Y. Liu and K. Xue, "An improved secure and efficient password and chaos-based two-party key agreement protocol," *Nonlinear Dynamics*, vol. 84, no. 2, pp. 549–557, 2016.
- [37] Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li, and A. Alelaiwi, "Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy," *Nonlinear Dynamics*, vol. 83, no. 4, pp. 2085–2101, 2016.
- [38] D. Wang, N. Wang, P. Wang, and S. Qing, "Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity," *Information Sciences*, vol. 321, pp. 162–178, 2015.
- [39] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.
- [40] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, p. 1, 2020.
- [41] GTI, "GTI security consideration for 5G smart city whitepaper," 2020, <http://www.gtigroup.org/Resources/rep/2020-03-16/14833.html>.
- [42] W. Zhou, X. T. Zhu, and X. Xia, "Study on 5G V2X communication security service," *Application of Electronic Technique*, vol. 45, no. 12, pp. 34–37, 2019.
- [43] J. P. Wang and G. Q. Liu, "High security level special industry 5G application security architecture," *Confidential Science and Technology*, vol. 1, pp. 16–21, 2019.
- [44] ETSI, *Multi-Access Edge Computing (MEC), Framework and Reference Architecture*, 2019.
- [45] M. He, J. Shen, G. W. Wu, and N. Fan, "Discussion on MEC security," *Mobile Communications*, vol. 43, no. 10, pp. 2–6, 2019.
- [46] X. J. Zhuang, B. Yang, X. Wang, and J. Peng, "Approach on mobile edge computing security," *Telecom Engineering Techniques and Standardization*, vol. 31, no. 12, pp. 38–43, 2018.
- [47] W. Zhou, "Research on 5G network slicing security technology," *Mobile Communications*, vol. 43, no. 10, pp. 38–42, 2019.
- [48] J. W. Liu, Y. R. Han, B. Liu, and B. Y. Yu, "Research on 5G network slicing security model," *Netinfo-Security*, vol. 20, no. 4, pp. 1–11, 2020.
- [49] S. Chen, W. T. Liang, N. Song, and K. Y. Fan, "Design of secure protection for 5G network slice," *Communication Technology*, vol. 52, no. 10, pp. 2499–2506, 2019.
- [50] S. Behrad, *Slice Specific Authentication and Access Control for 5G*, Doctoral School of the Polytechnic Institute of Paris, Doctoral Dissertation, 2020.
- [51] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, "Overview of 5G security challenges and solutions," *IEEE Communications Standards Magazine*, vol. 2, no. 1, pp. 36–43, 2018.
- [52] X. Duan and X. Wang, "Authentication handover and privacy protection in 5G hetnets using software-defined networking," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 28–35, 2015.
- [53] H. Luo, G. Wen, J. Su, and Z. Huang, "SLAP: succinct and lightweight authentication protocol for low-cost RFID system," *Wireless Networks*, vol. 24, no. 1, pp. 69–78, 2018.

- [54] J. Shao, D. Zhu, H. Jin, and R. Qiao, "A joint detection method for identifying pseudo base station based on abnormal access parameters," in *Proceedings of 2016 3rd International Conference on Engineering Technology and Application*, pp. 238–244, Kyoto, Japan, 2016.
- [55] China Mobile 5G Joint Innovation Center, *5G Slicing Security White Paper for Vertical Industries*, China Mobile 5G Joint Innovation Center, 2018.
- [56] Y.-J. Ku, D.-Y. Lin, C.-F. Lee et al., "5G radio access network design with the fog paradigm: confluence of communications and computing," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 46–52, 2017.
- [57] ISO/IEC 23188, *Information Technology – Cloud Computing – Edge Computing Landscape*, International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC), 2020.
- [58] ITU-T X5Gsec-netec, *Security Capabilities of Network Layer for 5G Edge Computing*.
- [59] ITU-T X5Gsec-ecs, *Security Framework for 5G Edge Computing Services*.
- [60] ETSI GS MEC 003 V111, *Mobile Edge Computing (MEC), Framework and Reference Architecture*, 2016.
- [61] ETSI GS MEC-IEG 004 V111, *Mobile-Edge Computing (MEC), Service Scenarios*, 2015.
- [62] ETSI GS MEC 002 V111, *Mobile Edge Computing (MEC), Technical Requirements*, 2016.
- [63] J. Zhang, Y. Zhao, B. Chen, F. Hu, and K. Zhu, "Research on edge computing data security and privacy protection," *Journal of Communications*, vol. 39, no. 3, pp. 1–21, 2018.
- [64] ETSI GS NFV-SEC 001 V111, *Network Functions Virtualisation (NFV), NFV Security; Problem Statement*, 2014.
- [65] ETSI GS NFV-SEC 003 V111, *Network Functions Virtualisation (NFV), NFV Security; Security and Trust Guidance*, 2014.
- [66] ETSI GS NFV-SEC 012 V311, "Network functions virtualisation (NFV) release 3," *Security; System Architecture Specification for Execution of Sensitive NFV Components*, European Telecommunications Standards Institute (ETSI), 2017.
- [67] ITU-T X1038, *Security Requirements and Reference Architecture for Software-Defined Networking*, International Telecommunications Union (ITU), 2019.
- [68] Y. Wang, W. Chu, S. Fields, C. Heinemann, and Z. Reiter, "Detection of intelligent intruders in wireless sensor networks," *Future Internet*, vol. 8, no. 4, p. 2, 2016.
- [69] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
- [70] B. Zhang, J. Yuan, Q. Qiu, X. J. Li, and F. Zhang, "Research on 5G security technology and development," in *Proceedings of "5G +" China Mobile Science and Technology Association*, pp. 1–5, Beijing, China, 2019.
- [71] N. Fan, G. Liu, and J. Shen, "Analysis of mobile network security for operators in the initial stage of 5G commercialization," *China Information Security*, no. 7, pp. 85–87, 2019.
- [72] China Mobile 5G Joint Innovation Center, *White Paper on 5G Security for the Medical Industry*, China Mobile 5G Joint Innovation Center, 2019.
- [73] A. K. Das, S. Zeadally, and M. Wazid, "Lightweight authentication protocols for wearable devices," *Computers & Electrical Engineering*, vol. 63, pp. 196–208, 2017.

## Research Article

# Two-Round Password-Based Authenticated Key Exchange from Lattices

Anqi Yin <sup>1</sup>, Yuanbo Guo <sup>1</sup>, Yuanming Song <sup>2</sup>, Tongzhou Qu <sup>1</sup>, and Chen Fang <sup>1</sup>

<sup>1</sup>Zhengzhou Institute of Information Science and Technology, Henan 450001, China

<sup>2</sup>School of EECS, Peking University, Beijing 100871, China

Correspondence should be addressed to Yuanbo Guo; guo\_yuanbo@126.com

Received 7 August 2020; Revised 7 September 2020; Accepted 30 September 2020; Published 14 December 2020

Academic Editor: Qi Jiang

Copyright © 2020 Anqi Yin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Password-based authenticated key exchange (PAKE) allows participants sharing low-entropy passwords to agree on cryptographically strong session keys over insecure networks. In this paper, we present two PAKE protocols from lattices in the two-party and three-party settings, respectively, which can resist quantum attacks and achieve mutual authentication. The protocols in this paper achieve two rounds of communication by carefully utilizing the splittable properties of the underlying primitive, a CCA (Chosen-Ciphertext Attack)-secure public key encryption (PKE) scheme with associated nonadaptive approximate smooth projection hash (NA-ASPH) system. Compared with other related protocols, the proposed two-round PAKE protocols have relatively less communication and computation overhead. In particular, the two-round 3PAKE is more practical in large-scale communication systems.

## 1. Introduction

Password-based authentication key exchange (PAKE) is theoretically fascinating, since it allows participants sharing short, low-entropy passwords to agree on cryptographically strong session keys over insecure networks [1, 2]. PAKE protocols are very practical as passwords are probably the most common and widely used authentication method [3–6], and password-based authentication can avoid the dependence on public key infrastructure and secure hardware; thereby, it improves the convenience of the system.

However, the use of shared short, low-entropy passwords will expose PAKE to greater security threats. This is because it must be ensured that the protocol is immune to off-line dictionary attacks, in which the adversary can exhaust all possible passwords to determine the correct one [7, 8]. Another observation is that an adversary can always succeed by guessing the password as the password dictionary is relatively small (usually polynomial in the security parameter), referred to as an on-line attack. The aim of PAKE is thus to limit the adversary to such an attack only [9].

The first successful password-based authenticated key exchange agreement methods were Encrypted Key Exchange methods described by Steven M. Bellovin and Michael Merritt in 1992 [10]. Initial PAKE protocols are generally based on “hybrid” models [11, 12], in which the clients need to store the public key of the server besides sharing a password with the server. The requirement of securely storing long, high-entropy public key is not friendly in multiserver environment. This motivates the study towards password-only protocols [13–16] where clients need to remember only a short password. Most PAKE protocols above provide only informal security arguments. Thus, Bellare et al. [17] and Boyko et al. [18] gave formal models of security of the password-only setting and proved security in the ideal cipher model and random oracle model, respectively. And then, Goldreich and Lindell [19] presented a provably secure password-only key exchange under standard cryptographic assumptions. Their work shows the possibility for password-based authentication under very weak assumptions, but the protocol itself is far from practical.

Later, many provably secure PAKE protocols based on various hardness assumptions were proposed. The research is mainly divided into two directions. The former is PAKE in the random oracle/ideal cipher model, which aims to achieve the highest possible levels of performance [20–22]. The latter is dedicated to seeking more efficient PAKE in the standard model [5, 23–25].

The first efficient PAKE protocol under standard model was proposed by Katz et al. [7]. They utilized CCA2 (Adaptive Chosen-Ciphertext Attack)-secure encryption system and corresponding smooth projection hash (SPH) function for key exchange to construct their scheme. Then, Gennaro and Lindell [9] abstracted their work and presented a corresponding PAKE framework, referred to as KOY/GL framework without mutual authentication. Based on KOY/GL framework, Jiang and Gong [26] showed a three-round PAKE supporting mutual authentication. Groce and Katz [5] then generalized the protocol by Jiang and Gong and gave a new PAKE framework in the common reference string model (CRS), referred to as JG/GK framework. Subsequently, based on the above two framework, a series of PAKE protocols [27–29] with different security are proposed for different application scenarios.

Most above schemes are two-party PAKE (2PAKE), requiring every two participants to share a password, which is not adaptable to large communication systems. In contrast, three-party PAKE (3PAKE) enables each client to share a password with the server for authentication, thereby avoiding the limitation of 2PAKE. Abdalla et al. [30] gave a general structure of the 3PAKE protocol for the first time. Subsequently, cryptographers designed a series of 3PAKE protocols with different efficiency and security [31, 32].

The security of most protocols above relies on traditional difficult problems (such as large integer factorization and discrete logarithm problems), so they cannot resist quantum attacks. However, the public-key cryptosystem based on the lattice assumption can resist quantum attacks. In addition, the operation on the lattice is matrix-vector multiplication which can be practically implemented by parallel computing. Therefore, the public-key cryptosystem from lattices will be more secure and efficient.

In lattice-based cryptosystem, the research on PAKE is relatively insufficient. In 2009, Katz et al. [33] presented the first lattice-based 2PAKE protocol based on KOY/GL framework. They proposed their protocol by constructing the first lattice-based CCA-secure encryption system and its corresponding approximate smooth projected hash (ASPH) function. Ding et al. [34] then applied the encryption system and ASPH function from Katz et al. [33] to JG/GK framework, and a more efficient protocol is given in the standard model.

Ye et al. [35] proposed the first 3PAKE protocol based on the JG/GK framework from lattices, and they proved its security under the standard model. This is a three-round protocol that implements explicit mutual authentication between the client and the server. In 2017, Xu et al. [36] proposed a provably secure 3PAKE protocol based on the R-LWE (ring learning with error) problem according to the idea of DH, but this protocol suffers from low efficiency.

Zhang et al. [1] applied a splittable public key encryption system to the KOY/GL framework and proposed a lattice-based PAKE, requiring only two-round communication, so it is more efficient. However, Zhang’s 2PAKE cannot be directly applied to the 3PAKE protocol, because another function is needed to compute the client-side information.

In this paper, we present efficient new constructions of 2PAKE and 3PAKE based on the learning with error (LWE) problem based on ideas of [1, 34, 35]. We then prove the security of the proposed protocols in the random oracle model. Significant security (resistance to quantum attacks) and efficiency improvements would also be obtained when basing the protocol on lattice assumption. Compared with the general structure [30], the new protocols reduce the number of communications, thereby improving efficiency. Our protocols also achieve mutual authentication between participants, so they can resist unpredictable on-line attacks. And the proposed two-round 3PAKE is adaptable to large-scale communication systems.

## 2. Preliminaries

**2.1. Notations.** We denote the logarithm with base 2 (resp., the natural logarithm) by  $\text{lb}$  (resp.,  $\text{log}$ ). Vectors are expressed in columns and bold lower-case letters (for example,  $\mathbf{x}$ ). A matrix is considered as a collection of column vectors and is represented by bold capital letter (such as  $\mathbf{X}$ ). We denote the concatenation of  $\mathbf{X}$  and  $\mathbf{Y}$  as  $(\mathbf{X}||\mathbf{Y})$ . Let  $x \leftarrow_r K$  denote the random sampling of variable  $x$  from the distribution  $K$ . For any string  $x, y \in \{0, 1\}^l$ ,  $\text{Ham}(x, y)$  represents the Hamming distance between  $x$  and  $y$ . Table 1 summarizes the description of other symbols used in this paper.

### 2.2. Security Model

**2.2.1. Participants.** Participants include honest clients  $A, B, D, \dots \in \mathcal{U}$ , malicious clients  $\mathcal{A}, \mathcal{M}, \dots \in \mathcal{E}$  and trusted server  $C \in \mathcal{S}$ . For simplicity, we assume that the server set  $\mathcal{S}$  contains only one element  $C$ . For each distinct  $A, B \in \text{Client}$ , assume that  $A$  and  $B$  share a long-term key called password  $\text{pw}_{AB}$ . We simply assume that  $\text{pw}_{AB}$  is independently and uniformly chosen at random from the password dictionary  $\mathcal{D}$ . But our proof of security extends to more general cases. In the following, we refer to honest clients directly as clients, and the malicious clients as adversaries.

Each participant can execute multiple protocols with different partners at the same time. We call the execution of a protocol an instance. Denote the instance  $i$  of client  $A$  and the instance  $j$  of server  $C$  by  $\Pi_A^i$ , and  $\Pi_C^j$ , respectively.

Each instance  $\Pi_A^i$  maintains a local state vector  $(\text{sid}_A^i, \text{pid}_A^i, \text{sk}_A^i, \text{acc}_A^i, \text{term}_A^i)$ , where  $\text{sid}_A^i$  represents the session ID, recording all messages sent and received by  $\Pi_A^i$  in order;  $\text{pid}_A^i$  ( $\text{pid}_A^i \neq A$ ) represents the partner ID, the participant with which  $\Pi_A^i$  believes it is interacting;  $\text{sk}_A^i$  denotes the session key of  $\Pi_A^i$ ;  $\text{acc}_A^i$  and  $\text{term}_A^i$  are Boolean variables, indicating whether the  $\Pi_A^i$  is accepted or terminated.

TABLE 1: Symbol description.

Symbol	Descriptions
$\kappa$	Security parameter
pw	Password
$\mathcal{D}$	Password dictionary
$ S $	The size of set S
$\epsilon$	A real number in $(0, 1/2)$ .
$\ell(\ell = \ell(\kappa))$	An integer related to $\kappa$
sk	Session key
sk <sub>pk</sub>	The corresponding private key of pk

Before giving a formal definition of adversarial abilities, we first define partnering, correctness, and freshness as follows.

*Partnering.* If (1)  $\text{sid}_A^i = \text{sid}_B^j \neq \perp$  and (2)  $\Pi_A^i = B$  and  $\Pi_B^j = A$ , instances  $\Pi_A^i$  and  $\Pi_B^j$  are partnered.

*Correctness.* We say that the protocol between partnered instances  $\Pi_A^i$  and  $\Pi_B^j$  is correct if they are both accepted and establish the same session key, that is,  $\text{acc}_A^i = \text{acc}_B^j = 1$  and  $\text{sk}_A^i = \text{sk}_B^j$ .

*Freshness.* If none of the following cases happens, the instance  $\Pi_A^i$  is fresh: (1) adversary  $\mathcal{A}$  have sent a Reveal query to  $\Pi_A^i$ ; (2)  $\Pi_A^i$  and  $\Pi_B^j$  have become partners and adversary  $\mathcal{A}$  has sent a Reveal query to  $\Pi_B^j$ . The definition of Reveal query will be given below.

**2.2.2. Adversarial Abilities.** It is assumed that the protocol is executed over a generally insecure network. Adversary  $\mathcal{A}$  can eavesdrop, intercept, inject, and tamper with messages among different participants.  $\mathcal{A}$  can also obtain the session key of the accepted instances. The following oracle queries model the adversarial abilities, that is, the adversary's interaction with various instances.

- (i) *Execute*  $(A, i, B, j)$  Query. The oracle models off-line attacks for passive adversaries. This oracle executes the protocol between the client instances  $\Pi_A^i$  and  $\Pi_B^j$ , and it updates the state vectors according to the specific protocol. And return the transcript of this execution to  $\mathcal{A}$ .
- (ii) *Send*  $(A, i, M)$ . This oracle sends the message M to the client instance  $\Pi_A^i$  to update the corresponding state vector appropriately. Finally, it returns the output message of  $\Pi_A^i$  to  $\mathcal{A}$ . This oracle models on-line attacks from active adversaries.
- (iii) *Reveal*  $(A, i)$ . This oracle returns the session key of the accepted instance  $\Pi_A^i$  to  $\mathcal{A}$ , thereby modelling the leakage of the session key. This oracle corresponds to an on-line attack from active adversaries.
- (iv) *Test*  $(A, i)$ . The oracle selects a random bit  $b \leftarrow_r \{0, 1\}$ . And if  $b = 1$ , the real session key of  $\Pi_A^i$

is returned to  $\mathcal{A}$ . Otherwise, it returns a uniform string of appropriate length. Note that  $\mathcal{A}$  can only query this oracle once, and  $\mathcal{A}$  is only allowed to query a fresh instance. This oracle is used to define security and does not model any adversarial capability in the real world.

**2.2.3. The Advantage of the Adversary.** The security of the protocol is defined by a security experiment: the adversary is allowed to send a series of queries above, but the Test query can only be sent once; and the experiment ends with  $\mathcal{A}$  outputting bit  $b'$ , a guess of  $b$ . Informally,  $\mathcal{A}$  succeeds if (1)  $b' = b$ , that is,  $\mathcal{A}$ 's guess is correct, representing that the session key is insecure, (2)  $\mathcal{A}$  makes the instance accepted but there is no corresponding partner, indicating that the protocol cannot achieve mutual authentication. Formally, we use Success to indicate the success of  $\mathcal{A}$ . The advantage of the adversary in attacking the protocol  $\Pi$  is defined as  $\text{Adv}_{\Pi, \mathcal{A}} \stackrel{\text{def}}{=} 2 \Pr[\text{Success}] - 1$ .

**2.2.4. Secure Protocol.** Since the size of the password dictionary  $\mathcal{D}$  is usually small, a PPT (Probabilistic Polynomial Time) adversary can always succeed by exhausting  $\mathcal{D}$  in an on-line attack. Therefore, informally, if on-line attack is the best attack method for all PPT adversaries, the PAKE protocol is secure. Formally, we give the following definition of the secure protocol.

*Definition 1.* (secure protocol). A protocol is a secure PAKE with mutual authentication if for all password in dictionary  $\mathcal{D}$  and for any PPT adversary making at most  $Q(\kappa)$  on-line attacks, it holds that  $\text{Adv}_{\Pi, \mathcal{A}}(\kappa) \leq Q(\kappa)/|\mathcal{D}| + \text{negl}(\kappa)$  for some negligible function  $\text{negl}(\bullet)$ .

### 2.3. Splittable Labeled PKE System from Lattices

**2.3.1. Splittable Labeled PKE.** Let the splittable labeled CCA-secure PKE be SPKE = (KeyGen, Enc, Dec). KeyGen is a key generation algorithm outputting the public and secret key pair (pk, sk). Enc is an encryption algorithm that returns  $c = (u, v) = \text{Enc}(\text{pk}, \text{label}, \text{pw})$ , where  $u = f(\text{pk}, \text{pw})$ ,  $v = g(\text{pk}, \text{label}, \text{pw})$ ,  $f$  and  $g$  are two different subfunctions that constitute SPKE. Dec is a decryption algorithm defined as  $\text{pw} \leftarrow \text{Dec}(\text{label}, \text{sk}, c)$ . For any  $v'$  and  $\text{label}' \in \{0, 1\}^*$ , under the random selection of sk and  $r$ , the probability that  $\text{Dec}(\text{sk}, \text{label}, (u, v')) \notin \{\perp, \text{pw}\}$  is negligible in  $\kappa$ .

The ‘‘splittable’’ attribute is also reflected in the security of the public-key cryptosystem. When proving the CCA security of the splittable cryptosystem, the challenge phase of the CCA game should be modified as follows: (1) the adversary  $\mathcal{M}$  first sends two plaintexts  $\text{pw}_0, \text{pw}_1 \in \mathcal{D}$  of equal length. (2) The challenger  $\mathcal{CL}$  randomly chooses  $b^* \leftarrow_r \{0, 1\}$  and  $r^* \leftarrow_r \{0, 1\}^*$ . Then,  $\mathcal{CL}$  computes  $u^* = f(\text{pk}, \text{pw}_{b^*}, r^*)$  and returns  $u^*$  to  $\mathcal{M}$ . (3) Upon receiving  $u^*$ , the adversary  $\mathcal{M}$  submits  $\text{label} \leftarrow \{0, 1\}^*$ . (4)  $\mathcal{CL}$  computes  $v^* = g(\text{pk}, \text{label}, \text{pw}_{b^*}, r^*)$  and returns  $v^*$  to  $\mathcal{M}$ .

*Definition 2.* (CCA security of SPKE). SPKE is a secure CCA-secure public-key encryption scheme if for any PPT

adversary, it holds that  $\text{Adv}_{\text{SPKE}\dots}^{\text{IND-CCA}}(\kappa) \leq \text{negl}(\kappa)$  for some negligible function  $\text{negl}(\cdot)$ .

In this paper, we denote the splittable labeled CCA-secure PKE based on LWE problem [1] by  $\Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$ , and we will use it to construct two-round PAKEs. The definitions of the cryptographic primitives (TrapGen, CRSGen, Prove, Verify, and Solve) on which  $\Sigma$  is based can be found in [1]. TrapGen is a trapdoor generation algorithm for generating public keys and corresponding trapdoors; CRSGen is a common reference string generator, usually implemented by hardware; the Proof/Verify algorithm is similar to the signature/verification algorithm to ensure the integrity of  $(A_0, A_1, u, v, \beta)$ ; Solve is a trapdoor solving algorithm corresponding to TrapGen.

**2.3.2. A Splittable Labeled PKE from Lattices [1].** Suppose  $n_1, n_2 \in \mathbb{Z}$  and prime  $q$  is polynomial with respect to the security parameter  $\kappa$ . Let  $n = n_1 + n_2 + 1$ ,  $m = O(n \log q) \in \mathbb{Z}$ .  $\alpha, \beta \in \mathbb{R}$  are the parameters of the systems. The splittable labeled PKE from lattices  $\Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$  is defined as follows:

**KeyGen**( $1^\kappa$ ): given security parameter  $\kappa$ , we have  $(A_0, R_0) \leftarrow \text{TrapGen}(1^n, 1^m, q)$ ,  $(A_1, R_1) \leftarrow \text{TrapGen}(1^n, 1^m, q)$ , and  $\text{CRS} \leftarrow \text{CRSGen}(1^\kappa)$ . And return  $(\text{pk}, \text{sk}) = ((A_0, A_1, \text{CRS}), R_0)$ .

**Enc**( $\text{pk}, \text{label}, \text{pw}$ ): given  $\text{pk} = (A_0, A_1, \text{CRS})$ ,  $\text{label} \leftarrow \{0, 1\}^*$ , and plaintext  $\text{pw} \in \mathcal{D}$ , choose  $s_0, s_1 \leftarrow_r \mathbb{Z}_q^{n_1}, e_0, e_1 \leftarrow_r D_{\mathbb{Z}^m, \alpha q}$ . Return the ciphertext  $C = (u, v, \pi)$ , where

$$u = A_0^T \begin{pmatrix} s_0 \\ 1 \\ \text{pw} \end{pmatrix} + e_0, v = A_1^T \begin{pmatrix} s_1 \\ 1 \\ \text{pw} \end{pmatrix} + e_1 \quad (1)$$

and  $\pi \leftarrow \text{Prove}(\text{CRS}, (A_0, A_1, u, v, \beta), (s_0, s_1, \text{pw}), \text{label})$ .

**Dec**( $\text{sk}, \text{label}, C$ ): given  $\text{sk} = R_0$ ,  $\text{label} \leftarrow \{0, 1\}^*$ , and the ciphertext  $C = (u, v, \pi)$ , if  $\text{Verify}(\text{CRS}, (A_0, A_1, u, v, \beta), \pi, \text{label}) = 0$ , return  $\perp$ . Otherwise, compute

$$t = \begin{pmatrix} s_0 \\ 1 \\ \text{pw} \end{pmatrix} \leftarrow \text{Solve}(A_0, R_0, C) \quad (2)$$

Finally, return  $\text{pw} \in \mathbb{Z}_q^{n_2}$ .

**2.4. Nonadaptive Approximate Smooth Projective Hash (NA-ASPH) System.** Based on the smooth projected Hash function [37], Katz et al. [33] proposed an Approximate Smooth Projective Hash (ASPH) function that can be used to construct a lattice-based PAKE protocol. In our application, we use a modified definition of ASPH [1] from Katz's, referred to as nonadaptive approximate smooth projection hash (NA-ASPH) function.

Suppose that  $\Sigma(\text{Gen}, \text{Enc}, \text{Dec})$  is a semantically secure PKE system from lattices. Assume that a valid ciphertext  $c = (u, v)$  can be easily parsed as the output of the function

pair  $(f, g)$ . We use KeyGen to generate a key pair  $(\text{pk}, \text{sk})$  and we use  $C_{\text{pk}}$  to represent the valid ciphertext space corresponding to the public key  $\text{pk}$ . Define

$$\begin{aligned} X &= \{(\text{label}, c, \text{pw}) \mid (\text{label}, c) \in C_{\text{pk}}, \text{pw} \in \mathcal{D}\} \\ L &= \{(\text{label}, c, \text{pw}) \mid \text{label} \in \{0, 1\}^*, c = \text{Enc}(\text{pk}, \text{label}, \text{pw}, r)\} \\ \bar{L} &= \{(\text{label}, c, \text{pw}) \mid \text{label} \in \{0, 1\}^*, \text{pw} = \text{Dec}(\text{sk}, \text{label}, c)\}. \end{aligned} \quad (3)$$

Based on  $\Sigma(\text{Gen}, \text{Enc}, \text{Dec})$ , we introduce the  $\epsilon$ -NA-ASPH function defined by the sampling algorithm, which outputs  $(K, \ell, \mathbb{H} = \{H_k : X \rightarrow \{0, 1\}^\ell\}, S, \alpha : K \rightarrow S)$  given the public key  $\text{pk}$  of  $\Sigma$  (where  $K$  is the hash key space and  $k \in K$ ,  $\alpha$  is the key projection function from  $K$  to  $S$ , and  $S$  is the projection key space; the domain and value range of the  $\epsilon$ -NA-ASPH are  $X$  and  $\{0, 1\}^\ell$ , respectively), such that

- (1) There exist efficient algorithms for sampling a hash key  $k \leftarrow_r K$ , computing  $H_k(x) = H_k(u, \text{pw})$  for all  $x = (\text{label}, (u, v), \text{pw}) \in X$  and computing  $s = \alpha(k)$  for all  $k \in K$
- (2) For all  $k \leftarrow_r K$ ,  $x = (\text{label}, (u, v), \text{pw}) \in L$  and randomness  $r$ , there are efficient algorithms for computing  $\text{Hash}(s, x, r) = \text{Hash}(s, (u, \text{pw}), r)$  given  $u = f(\text{pk}, \text{pw}, r)$  and  $v = g(\text{pk}, \text{label}, \text{pw}, r)$ .

The  $\epsilon$ -NA-ASPH has the following properties:

- (i) *Correctness.* For all  $x = (\text{label}, (u, v), \text{pw}) \in L$  and  $s = \alpha(k)$ , it holds that  $\Pr[\text{Ham}((H_k(u, \text{pw})), \text{Hash}(s, (u, \text{pw}), r)) \geq \epsilon] = \text{negl}(\kappa)$  for some negligible function  $\text{negl}(\cdot)$ .
- (ii) *Smoothness.* For any (even unbounded) function  $h: S \rightarrow X \setminus \bar{L}$ ,  $k \leftarrow_r K$ ,  $s = \alpha(k)$ ,  $x = h(s)$  and  $\gamma \leftarrow_r \{0, 1\}^\ell$ , the distributions  $(s, H_k(x))$  and  $(s, \gamma)$  are statistically indistinguishable in the security parameter  $\kappa$

NA-ASPH has three modifications compared with ASPH in [33]: (1) the projection function  $\alpha$  depends only on the hash key  $k$ ; (2)  $H_k(x) = H_k(u, \text{pw})$  is determined by the hash key  $k$ , the first part of the ciphertext  $c = (u, v)$  and the plaintext  $\text{pw}$ ; (3) for all  $x = h(s) \notin \bar{L}$ , the smoothness holds. The first modification here enables the protocol proposed to achieve two rounds of communication, and the latter two are prepared to prove the security of the proposed protocol.

### 3. A Two-Round 2PAKE Protocol

We now describe the proposed two-round 2PAKE, which is based on the protocol by Groce and Katz [9] and the splittable PKE scheme by Zhang and Yu [1].

**3.1. Primitives.** The primitives we use are the following: (1) a splittable labeled PKE scheme  $\Sigma(\text{Gen}, \text{Enc}, \text{Dec})$  with an associated  $\epsilon$ -NA-ASPH  $(K, \ell, \mathbb{H} = \{H_k : X \rightarrow \{0, 1\}^\ell\}, S, \alpha : K \rightarrow S)$ , where the scheme  $\Sigma$  can be divided into function

TABLE 2: An honest execution of the two-round 2PAKE protocol.

Client A	Two-round 2PAKE	Server C
$r_1 \leftarrow_r \{0, 1\}^*$ $k_1 \leftarrow_r K$ $s_1 = \alpha(k_1)$ $\text{label}_1 = A \  C \  s_1$ $u_1 = f(\text{pk}, \text{pw}_A, r_1)$ $v_1 = g(\text{pk}, \text{label}_1, \text{pw}_A, r_1)$	$\underline{A \  s_1 \  c_1 = (u_1, v_1)}$	$k_2 \leftarrow_r K, k_2^* \leftarrow_r K$ $s_2 = \alpha(k_2), s_2^* = \alpha(k_2^*)$ $r_j \  \tau_j \  \text{sk}_j \leftarrow H_{k_2^*}(u_1, \text{pw}_A)$ $u_2 = f(\text{pk}, \text{pw}_A, r_j)$ $\text{tk} = \text{Hash}(s_1, (u_2, \text{pw}_A), r_j) \oplus H_{k_2}(u_1, \text{pw}_A)$ $\Delta = \text{tk} \oplus \text{ECC}(H_{k_2^*}(u_1, \text{pw}_A))$ $\text{label}_2 =$ $C \  A \  s_1 \  s_2 \  s_2^* \  \Delta \  c_1$ $v_2 = g(\text{pk}, \text{label}_2, \text{pw}_A, r_j)$
$\text{tk}' = H_{k_1}(u_2, \text{pw}_A) \oplus \text{Hash}(s_2, (u_1, \text{pw}_A), r_1)$ $H' = \text{ECC}^{-1}(\text{tk}' \oplus \Delta)$ If $\text{Ham}(H', \text{Hash}(s_2^*, (u_1, \text{pw}_A), r_1)) \leq 2\epsilon/\ell$ $r_i \  \tau_i \  \text{sk}_i \leftarrow H'$ $\text{sk}_{AC} = \text{sk}_i$	$\underline{s_2 \  s_2^* \  \Delta \  c_2 = (u_2, v_2)}$	$\text{sk}_{CA} = \text{sk}_j$

pair  $(f, g)$ ; (2) error-correcting code  $\text{ECC} : \{0, 1\}^k \rightarrow \{0, 1\}^\ell$  and the corresponding decoding algorithm  $\text{ECC}^{-1} : \{0, 1\}^\ell \rightarrow \{0, 1\}^k$ . ECC can correct  $2\epsilon$ -fraction of errors. We assume that if  $\rho$  is sampled uniformly from  $\{0, 1\}^\ell$ ,  $\mu = \text{ECC}^{-1}(\rho)$  is uniformly distributed in  $\{0, 1\}^k$  provided that  $\mu \neq \perp$ .

**3.2. Initialization.** The proposed protocol requires the public key  $\text{pk}$  of the scheme  $\Sigma$ , also known as the common reference string (CRS). We want to emphasize that during the execution of the entire protocol, no participant needs to know the private key corresponding to the public key.

**3.3. Protocol Execution.** A high-level depiction of the two-round 2PAKE protocol is given in Table 2. We assume that the execution of the protocol is between client A and server C. Client A and server C share a password  $\text{pw}_A \in \mathcal{D}$ . When client A wants to initialize an authentication with the server C, A chooses a random tape  $r_1 \leftarrow_r \{0, 1\}^*$  for encryption and a hash key  $k_1 \leftarrow_r K$  for the NA-ASPH. Then, client A computes the projection key  $s_1 = \alpha(k_1)$ , sets  $\text{label}_1 = A \| C \| s_1$ . And A computes  $c_1 = (u_1, v_1) = \Sigma(\text{pk}, \text{label}_1, \text{pw}_A, r_1)$ , where  $u_1 = f(\text{pk}, \text{pw}_A, r_1)$  and  $v_1 = g(\text{pk}, \text{label}_1, \text{pw}_A, r_1)$ . Finally, client A sends the message  $(A \| s_1 \| c_1 = (u_1, v_1))$  to server C.

After receiving  $(A \| s_1 \| c_1)$  from client A, server C checks whether  $c_1$  is a valid ciphertext with respect to  $\text{pk}$  and  $\text{label}_1 = A \| C \| s_1$ . If not, C rejects and the protocol aborts. Otherwise, C chooses hash keys  $k_2 \leftarrow_r K$  and  $k_2^* \leftarrow_r K$ , and it computes projection keys  $s_2 = \alpha(k_2)$ ,  $s_2^* = \alpha(k_2^*)$ ,  $r_j \| \tau_j \| \text{sk}_j \leftarrow H_{k_2^*}(u_1, \text{pw}_A)$ ,  $u_2 = f(\text{pk}, \text{pw}_A, r_j)$ ,  $\text{tk} = \text{Hash}(s_1, (u_2, \text{pw}_A), r_j) \oplus H_{k_2}(u_1, \text{pw}_A)$ , and  $\Delta = \text{tk} \oplus \text{ECC}(H_{k_2^*}(u_1, \text{pw}_A))$ . Server C then sets  $\text{label}_2 = C \| A \| s_1 \| s_2 \| s_2^* \| \Delta$

and computes  $v_2 = g(\text{pk}, \text{label}_2, \text{pw}_A, r_j)$ . Finally, server C sends to client A the message  $(s_2 \| s_2^* \| \Delta \| c_2 = (u_2, v_2))$  and outputs  $\text{sk}_C = \text{sk}_j$ .

Upon receiving  $(s_2 \| s_2^* \| \Delta \| c_2)$  from server C, client A checks whether  $c_2$  is a valid ciphertext with respect to  $\text{pk}$  and  $\text{label}_2 = C \| A \| s_1 \| s_2 \| s_2^* \| \Delta \| c_1$ . If not, client A rejects and the protocol aborts. Otherwise, client A computes  $\text{tk}' = H_{k_1}(u_2, \text{pw}_A) \oplus \text{Hash}(s_2, (u_1, \text{pw}_A), r_1)$ ,  $H' = \text{ECC}^{-1}(\text{tk}' \oplus \Delta)$ . Then it checks whether the Hamming distance between  $H'$  and  $\text{Hash}(s_2^*, (u_1, \text{pw}_A), r_1)$  is less than  $2\epsilon/\ell$ . If not, client A rejects and the protocol aborts. Otherwise, client A sets  $r_i \| \tau_i \| \text{sk}_i \leftarrow H'$  and outputs  $\text{sk}_A = \text{sk}_i$ .

**3.4. Correctness.** After honestly executing the protocol, participants can obtain different session keys with negligible probability. First, according to the smoothness of NA-ASPH, we can conclude that both  $H_{k_1}(u_2, \text{pw}_A) \oplus \text{Hash}(s_2, (u_1,$



$\text{pw}_A, r_1)$  and  $\text{Hash}(s_1, (u_2, \text{pw}_A), r_j) \oplus H_{k_2}(u_1, \text{pw}_A)$  have at most  $\epsilon$ -fraction of nonzeros. Therefore,  $\text{tk} \oplus \text{tk}'$  has at most  $2\epsilon$ -fraction of nonzeros. Then, we can obtain that  $H' = \text{ECC}^{-1}(\text{tk}' \oplus \Delta) = \text{ECC}^{-1}(\text{tk}' \oplus \text{tk} \oplus \text{ECC}(H_{k_2}(u_1, \text{pw}_A))) = H_{k_2}(u_1, \text{pw}_A)$  as we assume that ECC can correct  $2\epsilon$ -fraction of errors. Second, we verify the validity of  $H_{k_2}(u_1, \text{pw}_A)$  by checking whether the Hamming distance between  $H'$  and  $\text{Hash}(s_2^*, (u_1, \text{pw}_A), r_1)$  is less than  $2\epsilon/\ell$ . If it is the case, it holds that  $\text{sk}_i = \text{sk}_j$ . This completes the correctness argument.

**3.5. Security.** We now show that the above two-round 2PAKE is secure through the proof of the following theorem.

**Theorem 1.** *If  $\Sigma(\text{Gen}, \text{Enc}, \text{Dec})$  is a splittable CCA-secure PKE scheme associated with an  $\epsilon$ -NA-ASPH  $(K, \ell, \mathbb{H} = \{H_k : X \rightarrow \{0, 1\}^\ell\}, S, \alpha : K \rightarrow S)$  and  $\text{ECC} : \{0, 1\}^k \rightarrow \{0, 1\}^\ell$  is an error-correcting code which can correct  $2\epsilon$ -fraction of errors, then the protocol in Table 2 is a secure PAKE protocol.*

*Proof.* Suppose  $\mathcal{A}$  is a PPT attacker targeting this protocol. We estimate the advantage of adversary  $\mathcal{A}$  through a series of experiments  $\mathcal{T}_0, \mathcal{T}_1, \mathcal{T}_2, \dots$ , where  $\mathcal{T}_0$  represents the experiment in the real protocol. By analyzing the difference of adversary's advantage between two adjacent experiments and defining the adversary's advantage in the final experiment, we can finally get the adversary's advantage in experiment  $\mathcal{T}_0$ , that is, the adversary's advantage when attacking the real protocol. In experiment  $\mathcal{T}_i$ , the event  $\text{Success}_i$  indicates that adversary  $\mathcal{A}$  succeeds, and the adversary's advantage is defined as  $\text{Adv}_{\mathcal{A}, i}(\kappa) = 2 \Pr[\text{Success}_i] - 1$ .

**Experiment  $\mathcal{T}_0$ .** This experiment corresponds to the security experiment of the real protocol. Attackers can send all queries according to the regulations of the secure model, and the instance being queried will respond according to the actual protocol specifications.

**Experiment  $\mathcal{T}_1$ .** We change the simulation method of Execute query. The only difference from the experiment  $\mathcal{T}_0$  is that the calculation method of  $\text{tk}'$  is changed to  $\text{tk}' = H_{k_1}(u_2, \text{pw}_A) \oplus H_{k_2}(u_1, \text{pw}_A)$ .

**Lemma 1.** *If  $(K, \ell, \mathbb{H} = \{H_k : X \rightarrow \{0, 1\}^\ell\}, S, \alpha : K \rightarrow S)$  is an  $\epsilon$ -NA-ASPH, and  $\text{ECC} : \{0, 1\}^k \rightarrow \{0, 1\}^\ell$  is an error-correcting code which can correct  $2\epsilon$ -fraction of errors, then  $|\text{Adv}_{\mathcal{A}, 1}(\kappa) - \text{Adv}_{\mathcal{A}, 0}(\kappa)| \leq \text{negl}(\kappa)$ .*

*Proof.* Since the simulator knows  $k_1$  and  $k_2$ , it is easy to know that Lemma 1 holds according to the approximate correctness of NA-ASPH and the correctness of ECC.

**Experiment  $\mathcal{T}_2$ .** Compared with experiment  $\mathcal{T}_1$ , we modify the response to the Execute query as shown below. The ciphertext  $c_1$  is replaced by the encryption of the illegal

password  $\text{pw}'_A \notin \mathcal{D}$ , and  $\text{tk}'$  calculated by the client A is forced to be equal to the  $\text{tk}$  calculated by the server C, and other calculations remain unchanged.

**Lemma 2.** *If  $\Sigma(\text{Gen}, \text{Enc}, \text{Dec})$  is a CCA-secure PKE scheme, then  $|\text{Adv}_{\mathcal{A}, 2}(\kappa) - \text{Adv}_{\mathcal{A}, 1}(\kappa)| \leq \text{negl}(\kappa)$ .*

*Proof.* We use standard hybrid argument to analyze the impact of replacing  $\text{pw}_A$  with  $\text{pw}'_A$  on the adversary's advantage. We set the number of queries to  $q_{\text{exe}}$  and define a series of intermediate experiments. The first  $\eta$  queries in the experiment are same as those in  $\mathcal{T}_2$ , the remaining  $(q_{\text{exe}} - \eta)$  queries are same as those in  $\mathcal{T}_1$ . The Send query conforms to the security model. It can be seen that experiments  $\mathcal{T}_1^{(0)}$  and  $F_1^{(q_{\text{exe}})}$  are completely consistent with the experiments  $\mathcal{T}_1$  and  $\mathcal{T}_2$ , respectively. If the lemma is not true, that is, the difference of  $\mathcal{A}$ 's advantage between experiments  $\mathcal{T}_1$  and  $\mathcal{T}_2$  is not negligible, there must be some  $\eta$  such that the difference of  $\mathcal{A}$ 's advantage between  $\mathcal{T}_1^{(\eta-1)}$  and  $\mathcal{T}_1^{(\eta)}$  cannot be ignored. Then, we can construct an attacker  $\mathcal{M}$  for the security of the encryption system  $\Sigma$  so that it can successfully attack  $\Sigma$  with nonnegligible probability.

We now construct an adversary  $\mathcal{M}$  who attacks the CCA-secure PKE scheme  $\Sigma$  in the following way: given the public key  $\text{pk}$ , the adversary  $\mathcal{M}$  simulates the entire experiment for  $\mathcal{A}$  according to the experiment  $\mathcal{T}_1^{(\eta)}$ , including selecting random passwords for the participants and selecting the random bit  $b$  for  $\mathcal{A}$  in the Test query. When answering the Execute query,  $\mathcal{M}$  sends  $(\text{pw}_A, \text{pw}'_A)$  as its challenge plaintext pair to  $\mathcal{M}$ 's own challenger. After receiving the challenge ciphertext  $c'_1$ ,  $\mathcal{M}$  replaces  $c_1$  with  $c'_1$  in the Execute query. Finally,  $\mathcal{M}$  checks whether  $\mathcal{A}$  guesses the random bit in the Test query. If  $\mathcal{A}$  succeeds,  $\mathcal{M}$  outputs 1; otherwise,  $\mathcal{M}$  outputs 0.

Let  $\text{Event}_{\Sigma}^{\text{pw}_A}(\mathcal{M})$  denote that  $\mathcal{M}$  obtains the challenge ciphertext of the real password  $\text{pw}_A$  and outputs 1 at the end of the experiment. Let  $\text{Event}_{\Sigma}^{\text{pw}'_A}(\mathcal{M})$  indicate that  $\mathcal{M}$  obtains the challenge ciphertext of the invalid password  $\text{pw}'_A$  and outputs 1 at the end of the experiment. For the  $\eta^{\text{th}}$  query, that is,  $\mathcal{M}$  gets the challenge ciphertext of the real password  $\text{pw}_A$ , the environment provided by  $\mathcal{M}$  for the protocol attacker  $\mathcal{A}$  is the same as experiment  $\mathcal{T}_1^{(\eta-1)}$ . Therefore, in experiment  $\mathcal{T}_1^{(\eta-1)}$ , the probability that  $\mathcal{M}$  outputs 1 is exactly the same as  $\mathcal{A}$ 's success probability ( $\Pr[\text{Success}_{\mathcal{T}_1^{(\eta-1)}}]$ ), i.e.,  $\Pr[\text{Event}_{\Sigma}^{\text{pw}_A}(\mathcal{M}) = 1] = \Pr[\text{Success}_{\mathcal{T}_1^{(\eta-1)}}]$ . Similarly, when  $\mathcal{M}$  gets the challenge ciphertext of the invalid password  $\text{pw}'_A$ , the probability that  $\mathcal{M}$  outputs 1 is the probability that  $\mathcal{A}$  succeeds ( $\Pr[\text{Success}_{\mathcal{T}_1^{(\eta)}}]$ ) in attacking the protocol in experiment  $\mathcal{T}_1^{(\eta)}$ , namely  $\Pr[\text{Event}_{\Sigma}^{\text{pw}'_A}(\mathcal{M}) = 1] = \Pr[\text{Success}_{\mathcal{T}_1^{(\eta)}}]$ . Let  $\text{Adv}_{\Sigma, \mathcal{M}}^{\text{IND-CCA}}(\kappa)$  be  $\mathcal{M}$ 's advantage in attacking the encryption system  $\Sigma$ , then

$$\begin{aligned}
& \left| \text{Adv}_{\mathcal{T}_1^{(\eta)}}(\kappa) - \text{Adv}_{\mathcal{T}_1^{(\eta-1)}}(\kappa) \right| \\
&= 2 \left| \Pr \left[ \text{Success}_{\mathcal{T}_1^{(\eta)}} \right] - \Pr \left[ \text{Success}_{\mathcal{T}_1^{(\eta-1)}} \right] \right| \\
&= 2 \left| \Pr \left[ \text{Event}_{\Sigma}^{\text{PWA}}(\mathcal{M}) = 1 \right] - \Pr \left[ \text{Event}_{\Sigma}^{\text{PWA}}(\mathcal{M}) = 1 \right] \right| \\
&= 2 \text{Adv}_{\Sigma, \mathcal{M}}^{\text{IND-CCA}}(\kappa).
\end{aligned} \tag{4}$$

According to the CCA security of encryption system  $\Sigma$ , the lemma holds. We emphasize that only the CPA security of  $\Sigma$  is actually used here.

Experiment  $\mathcal{T}_3$ . We change the response to the Execute query: (1) change the calculation method of tk to  $\text{tk} = H_{k_1}(u_2, \text{pw}_A) \oplus H_{k_2}(u_1, \text{pw}_A)$ . (2) Replace the ciphertext  $c_2$  with the encryption of an illegal password  $\text{pw}'_A \notin \mathcal{D}$ .

**Lemma 3.** *If  $\Sigma(\text{Gen}, \text{Enc}, \text{Dec})$  is a splittable CCA-secure PKE scheme associated with and  $\epsilon$ -NA-ASPH, and  $\text{ECC} : \{0, 1\}^k \rightarrow \{0, 1\}^\ell$  is an error-correcting code which can correct  $2\epsilon$ -fraction of errors, then  $|\text{Adv}_{\mathcal{A}, 3}(\kappa) - \text{Adv}_{\mathcal{A}, 2}(\kappa)| \leq \text{negl}(\kappa)$ .*

*Proof.* This lemma is shown through a series of experiments similar to  $\mathcal{T}_1$ ,  $\mathcal{T}_2$ , and  $\mathcal{T}_3$ . In addition, this experiment utilizes the modified CCA security experiment shown in Section 2.3 instead of the standard CCA security experiment.

Experiment  $\mathcal{T}_4$ . We continue to modify the response to the Execute query as follows. We set  $r_j || \tau_j || \text{sk}_j$  to a random string of the appropriate length and the  $r_i || \tau_i || \text{sk}_i$  calculated by client A to be equal to the  $r_j || \tau_j || \text{sk}_j$  calculated by server C.

**Lemma 4.** *If  $(K, \ell, \mathbb{H} = \{H_k : X \rightarrow \{0, 1\}^\ell\}, S, \alpha : K \rightarrow S)$  is an  $\epsilon$ -NA-ASPH, then  $|\text{Adv}_{\mathcal{A}, 4}(\kappa) - \text{Adv}_{\mathcal{A}, 3}(\kappa)| \leq \text{negl}(\kappa)$ .*

*Proof.* This comes from the smoothness of NA-ASPH, because when responding to an Execute query in  $\mathcal{T}_3$ , the hash function  $H_k$  is always applied to  $\text{pw}'_A \notin \mathcal{D}_n$ , so even if  $s$  is given, the output is statistically close to uniform. In addition, in  $\mathcal{T}_3$  and  $\mathcal{T}_4$  the string  $r_i || \tau_i || \text{sk}_i$  used by the client is equal to the string  $r_j || \tau_j || \text{sk}_j$  computed by the server.

Note that the Execute query in  $\mathcal{T}_4$  will generate a random session key and random transcripts, which have nothing to do with the actual password of any participant. In the following experiment, we begin to modify the responses to the Send queries. Let  $\text{Send}_0(A, i, C)$  represent the “start” message, which enables the client instance  $\Pi_A^i$  to initiate authentication with the server S. Note that when calculating the number of communication rounds, we ignore the “start” message like other related research. Let  $\text{Send}_1(C, j, \text{msg1} = (s_1 || c_1 = (u_1, v_1)))$  represent the first message of the protocol sent to the server instance  $\Pi_C^j$ . Let  $\text{Send}_2(A, i, \text{msg2} = (s_2 || s_2^* || \Delta || c_2 = (u_1, v_1)))$  denote the second message of the protocol sent to the client instance  $\Pi_A^i$ . We also record the secret key  $\text{sk}_{\text{pk}}$ , corresponding to the public key in the generated CRS.

Now, we make some explanations for msg1 and msg2. The output of  $\text{send}_0$  oracle or the input of  $\text{send}_1$  oracle are msg1. Similarly, msg2 may be the output of  $\text{send}_1$  oracle or the output of  $\text{send}_2$  oracle. If msg1/msg2 is output by a previous  $\text{send}_0/\text{send}_1$  oracle, then we call msg1/msg2 oracle-generated.

Experiment  $\mathcal{T}_5$ . In experiment  $\mathcal{T}_5$ , we change the response to  $\text{send}_1$  queries. If msg1 is oracle-generated, the experiment is the same as  $\mathcal{T}_4$ . Otherwise, we set  $\text{label}_1 = A || C || s_1$ . We check the validity of  $c_1$  according to  $\text{label}_1$  and  $\text{pk}$ .

- (i) If  $c_1$  is invalid, the experiment just aborts as the real protocol
- (ii) Else, we can get  $\text{pw}_A^{\text{ad}}$  by decrypting  $c_1$ , because we have  $\text{sk}_{\text{pk}}$ . If  $\text{pw}_A^{\text{ad}} = \text{pw}_A$ , we just declare that adversary  $\mathcal{A}$  succeeds, and the experiment is terminated. If  $\text{pw}_A^{\text{ad}} \neq \text{pw}_A$ , we set tk and  $H_{k_2}^*(u_1, \text{pw}_A)$  computed by the server as random tapes of the appropriate length

**Lemma 5.** *If  $(K, \ell, \mathbb{H} = \{H_k : X \rightarrow \{0, 1\}^\ell\}, S, \alpha : K \rightarrow S)$  is an  $\epsilon$ -NA-ASPH, then  $\text{Adv}_{\mathcal{A}, 4}(\kappa) \leq \text{Adv}_{\mathcal{A}, 5}(\kappa) + \text{negl}(\kappa)$ .*

*Proof.* In the actual protocol, server C simply refuses, and the protocol terminates when  $c_1$  is invalid. Therefore, if msg1 is oracle-generated, or msg1 is not oracle-generated and  $c_1$  is invalid, then experiment  $\mathcal{T}_5$  is consistent with experiment  $\mathcal{T}_4$ . Now, we only need to consider the case where msg1 is not oracle-generated and  $c_1$  is valid.

- (i) If  $\text{pw}_A^{\text{ad}} = \text{pw}_A$ , adversary  $\mathcal{A}$  succeeds. Note that this only improves the adversary’s advantage
- (ii) If  $\text{pw}_A^{\text{ad}} \neq \text{pw}_A$ , tk and  $H_{k_2}^*(u_1, \text{pw}_A)$  computed by the server are both set to random tapes. From the view of adversary  $\mathcal{A}$ , there is no difference between these changes. First, as  $(c_1, \text{pw}_A^{\text{ad}}) \notin \bar{L}$ , in the view of  $\mathcal{A}$ , both  $\text{tk} = \text{Hash}(s_1, (u_2, \text{pw}_A), r_j) \oplus H_{k_2}$

$(u_1, \text{pw}_A)$  and  $\Delta = \text{tk} \oplus \text{ECC}(H_{k_2}^*(u_1, \text{pw}_A^{\text{ad}}))$  are statistically indistinguishable from random uniform distribution. This can be derived directly from the smoothness of NA-ASPH. Similarly, from the view of  $\mathcal{A}$ ,  $r_j || \tau_j || \text{sk}_j = H_{k_2}^*(u_1, \text{pw}_A^{\text{ad}})$  is also statistically indistinguishable from random uniform distribution. Therefore,  $\text{pw}_A^{\text{ad}} \neq \text{pw}_A$  only introduces a negligible difference in experiment  $\mathcal{T}_5$ .

Finally, we obtain that  $\text{Adv}_{\mathcal{A}, 4}(\kappa) \leq \text{Adv}_{\mathcal{A}, 5}(\kappa) + \text{negl}(\kappa)$ .

Experiment  $\mathcal{T}_6$ . In experiment  $\mathcal{T}_6$ , let msg1 be the output from a previous  $\text{Send}_0$  query  $(A, i, C)$  (note that such a query must exist).  $\text{Send}_2$  query is handled as follows: If msg2 is oracle-generated by a previous  $\text{Send}_1$  query, the experiment is similar to  $\mathcal{T}_5$  except for (1) computing  $\text{tk}'$  as  $\text{tk}' = H_{k_1}(u_2, \text{pw}_A) \oplus H_{k_2}(u_1, \text{pw}_A)$  and (2) setting  $r_i || \tau_i || \text{sk}_i$

$= r_j || \tau_j || sk_j$ . Otherwise, we set  $label_2 = C || A || s_1 || s_2 || s_2^* || \Delta || c_1$ . We check the validity of  $c_2$  according to  $label_2$  and  $pk$ .

- (i) If  $c_2$  is invalid, the experiment just aborts as the real protocol
- (ii) Else, we can obtain  $pw_A^{ad}$  similar to  $\mathcal{T}_5$ . If  $pw_A^{ad} = pw_A$ , we declare that  $\mathcal{A}$  succeeds and the experiment terminates. Otherwise, if  $\Pi_A^I$  is accepted, let  $r_i || \tau_i || sk_i$  to be a random tape of appropriate length

**Lemma 6.** *If  $(K, \ell, \mathbb{H} = \{H_k : X \rightarrow \{0, 1\}^\ell\}, S, \alpha : K \rightarrow S)$  is an  $\epsilon$ -NA-ASPH, and  $ECC : \{0, 1\}^k \rightarrow \{0, 1\}^\ell$  is an error-correcting code which can correct  $2\epsilon$ -fraction of errors, then  $Adv_{\mathcal{A},5}(\kappa) \leq Adv_{\mathcal{A},6}(\kappa) + negl(\kappa)$ .*

*Proof.* We prove different situations separately. First, if both  $msg1$  and  $msg2$  are oracle-generated, the simulator will know the hash keys  $k_1$  and  $k_2$ . According to the smoothness of the NA-ASPH, the changes in computing  $tk'$  and  $r_i || \tau_i || sk_i$  are just conceptual (in this case, it holds that  $r_i || \tau_i || sk_i = r_j || \tau_j || sk_j$  in both  $\mathcal{T}_5$  and  $\mathcal{T}_6$ ). Second, if  $msg2$  is not oracle-generated, the simulator sets  $label_2 = C || A || s_1 || s_2 || s_2^* || \Delta || c_1$  and then uses  $label_2$  and  $pk$  to check whether  $c_2$  is valid. If not,  $\mathcal{T}_5$  and  $\mathcal{T}_6$  are the same as the real protocol. Otherwise, the simulator uses  $sk_{pk}$  to decrypt  $c_2$  and obtains  $pw_A^{ad}$ .

- (i) If  $pw_A^{ad} = pw_A$ , adversary  $\mathcal{A}$  succeeds. Note that this just improves the adversary advantage
- (ii) If  $pw_A^{ad} \neq pw_A$ , according to section 2.4, ( $label_2, c_2, pw_A^{ad}$ ) does not belong to  $\bar{L}$ . Then, by the smoothness of NA-ASPH,  $H_{k_1}(u_2, pw_A)$  and thus  $tk' = H_{k_1}(u_2, pw_A) \oplus H_{k_2}(u_1, pw_A)$  are both statistically close to uniform over  $\{0, 1\}^\ell$ . Furthermore, we have  $r_i || \tau_i || sk_i (r_i || \tau_i || sk_i \leftarrow H' = ECC^{-1}(tk' \oplus \Delta))$  is statistically close to uniform over  $\{0, 1\}^k$ . Therefore, the modifications of  $pw_A^{ad} \neq pw_A$  bring a negligible statistical difference. Note that the output of  $ECC^{-1}(tk' \oplus \Delta)$  may be  $\perp$ . In this case, client  $\mathcal{A}$  rejects

Experiment  $\mathcal{T}_7$ . Compared with experiment  $\mathcal{T}_6$ , we modify the response to a  $Send_0$  query. The only difference is that we use  $pw_A' \notin \mathcal{D}$  to compute  $c_1$ .

**Lemma 7.** *If  $\Sigma(Gen, Enc, Dec)$  is a CCA-secure PKE scheme, then  $|Adv_{\mathcal{A},7}(\kappa) - Adv_{\mathcal{A},6}(\kappa)| \leq negl(\kappa)$ .*

*Proof.* We analyze the impact of replacing  $pw_A$  with  $pw_A' \notin \mathcal{D}$  on the adversary's advantage similar to  $\mathcal{T}_2$ . But for the sake of simplicity, we consider that  $\mathcal{A}$  only executes a single  $Send_0$  query. The correctness still holds according to standard hybrid argument. If the lemma is not true, that is, the difference of  $\mathcal{A}$ 's advantage between experiments  $\mathcal{T}_6$  and  $\mathcal{T}_7$  is not negligible, then an attacker  $\mathcal{M}$  can be constructed for

the security experiment of the encryption system  $\Sigma$ , which can successfully attack  $\Sigma$  with nonnegligible probability.

We now construct an adversary  $\mathcal{M}$  who attacks the CCA-secure PKE scheme  $\Sigma$  in the following way: given the public key  $pk$ , the adversary  $\mathcal{M}$  simulates the entire experiment for  $\mathcal{A}$  according to experiment  $\mathcal{T}_7$ , including selecting random passwords for the participants and selecting the random bit  $b$  for  $\mathcal{A}$  in the Test query. When answering the  $Send_0$  query,  $\mathcal{M}$  will send  $(pw_A, pw_A')$  as its challenge plaintext pair to  $\mathcal{M}$ 's own challenger. After receiving the challenge ciphertext  $c_1'$ ,  $\mathcal{M}$  replaces  $c_1$  with  $c_1'$  in the  $Send_0$  query. Finally,  $\mathcal{M}$  checks whether  $\mathcal{A}$  guesses the random bit in the Test query. If  $\mathcal{A}$  succeeds,  $\mathcal{M}$  outputs 1; otherwise,  $\mathcal{M}$  outputs 0.

Let  $Event_{\Sigma}^{pw_A}(\mathcal{M})$  denote that  $\mathcal{M}$  gets the challenge ciphertext of the real password  $pw_A$  and outputs 1 at the end of the experiment. Let  $Event_{\Sigma}^{pw_A'}(\mathcal{M})$  represent that  $\mathcal{M}$  gets the challenge ciphertext of the invalid password  $pw_A'$  and outputs 1 at the end of the experiment. If  $\mathcal{M}$  gets the challenge ciphertext of the real password  $pw_A$ , the environment provided by  $\mathcal{M}$  for the protocol adversary  $\mathcal{A}$  is the same as experiment  $\mathcal{T}_6$ . Therefore, the probability that  $\mathcal{M}$  outputs 1 is exactly the same as  $\mathcal{A}$ 's success probability ( $\Pr[\text{Success}_{\mathcal{T}_6}]$ ) in experiment  $\mathcal{T}_6$ , i.e.,  $\Pr[Event_{\Sigma}^{pw_A}(\mathcal{M}) = 1] = \Pr[\text{Success}_{\mathcal{T}_6}]$ . Similarly,  $\Pr[Event_{\Sigma}^{pw_A'}(\mathcal{M}) = 1] = \Pr[\text{Success}_{\mathcal{T}_7}]$ . Let  $Adv_{\Sigma, \mathcal{M}}^{\text{IND-CCA}}(\kappa)$  be  $\mathcal{M}$ 's advantage in attacking the encryption system  $\Sigma$ , then

$$\begin{aligned} |\text{Adv}_{\mathcal{T}_7}^{(\eta)}(\kappa) - \text{Adv}_{\mathcal{T}_6}^{(\eta)}(\kappa)| &= 2 \left| \Pr[\text{Success}_{\mathcal{T}_7}] - \Pr[\text{Success}_{\mathcal{T}_6}] \right| \\ &= 2 \left| \Pr[Event_{\Sigma}^{pw_A'}(\mathcal{M}) = 1] \right. \\ &\quad \left. - \Pr[Event_{\Sigma}^{pw_A}(\mathcal{M}) = 1] \right| \\ &= 2 \text{Adv}_{\Sigma, \mathcal{M}}^{\text{IND-CCA}}(\kappa). \end{aligned} \tag{4}$$

According to the CCA security of the encryption system  $\Sigma$ , the lemma holds.

Experiment  $\mathcal{T}_8$ . Experiment  $\mathcal{T}_8$  is similar to  $\mathcal{T}_7$  except that if  $msg1$  is oracle-generated: (1)  $tk$  computed by the server is set to be  $H_{k_1}(u_2, pw_A) \oplus H_{k_2}(u_1, pw_A)$ ; (2) a random string  $r_j || \tau_j || sk_j$  of appropriate length is set for  $\Pi_C^j$ .

**Lemma 8.** *If  $\Sigma(Gen, Enc, Dec)$  is a splittable CCA-secure PKE scheme associated with an  $\epsilon$ -NA-ASPH  $(K, \ell, \mathbb{H} = \{H_k : X \rightarrow \{0, 1\}^\ell\}, S, \alpha : K \rightarrow S)$ , and  $ECC : \{0, 1\}^k \rightarrow \{0, 1\}^\ell$  is an error-correcting code which can correct  $2\epsilon$ -fraction of errors, then  $|Adv_{\mathcal{A},8}(\kappa) - Adv_{\mathcal{A},7}(\kappa)| \leq negl(\kappa)$ .*

*Proof.* First, if  $msg1$  is oracle-generated, the simulator has hash keys  $k_1$  and  $k_2$ ; thus, it can compute  $tk = H_{k_1}(u_2, pw_A) \oplus H_{k_2}(u_1, pw_A)$ . Secondly, since the ciphertext  $c_1$  is the encryption of  $pw_A' \notin \mathcal{D}$ ,  $H_{k_2}(u_1, pw_A)$  and  $tk$  are statistically close to uniform. Similarly,  $r_j || \tau_j || sk_j (r_j || \tau_j || sk_j \leftarrow H_{k_2}^*(u_1, pw_A))$  is also statistically close to uniform. Thus, we have that

the modifications here introduce only a statistically negligible difference. Therefore,  $|\text{Adv}_{\mathcal{A},8}(\kappa) - \text{Adv}_{\mathcal{A},7}(\kappa)| \leq \text{negl}(\kappa)$ .

Experiment  $\mathcal{T}_9$ . For the final experiment, we again modify the response to the  $\text{Send}_1$  queries. If  $\text{msg}_1$  is oracle-generated, the ciphertext  $c_2$  is now computed as the encryption of  $\text{pw}'_A \notin \mathcal{D}$ .

**Lemma 9.** *If  $\Sigma(\text{Gen}, \text{Enc}, \text{Dec})$  is a splittable CCA-secure PKE scheme, then  $|\text{Adv}_{\mathcal{A},9}(\kappa) - \text{Adv}_{\mathcal{A},8}(\kappa)| \leq \text{negl}(\kappa)$ .*

*Proof.* We analyze the impact of replacing  $\text{pw}_A$  with  $\text{pw}'_A \notin \mathcal{D}$  on the adversary's advantage. We consider that  $\mathcal{A}$  only executes a single  $\text{Send}_1$  query similar to  $\mathcal{T}_7$ . Now, we show that if any PPT adversary  $\mathcal{A}$  can distinguish these two experiments, then we can construct an attacker  $\mathcal{M}$  breaking the CCA security experiment (in Section 2.3) of the CCA encryption system  $\Sigma$  with a nonnegligible probability.

We now construct an adversary  $\mathcal{M}$  interacting with  $\mathcal{A}$  in  $\mathcal{T}_8$  to attack the CCA-secure PKE scheme  $\Sigma$  in the following way: given the public key  $\text{pk}$ , the adversary  $\mathcal{M}$  simulates the entire experiment, including selecting random passwords for the participants and selecting the random bit  $b$  for  $\mathcal{A}$  in the Test query. When answering the  $\text{Send}_1$  query,  $\mathcal{M}$  will send  $(\text{pw}_A, \text{pw}'_A)$  as its challenge plaintext pair to  $\mathcal{M}$ 's own challenger. After receiving the challenge ciphertext  $c'_1$ ,  $\mathcal{M}$  replaces  $c_1$  with  $c'_1$  in the  $\text{Send}_1$  query. When  $\mathcal{M}$  needs to decrypt some valid ciphertext  $c'_2$ , it will send  $(\text{label}'_2, c'_2)$  to its own challenger to obtain the corresponding  $\text{pw}'_A$ . Finally,  $\mathcal{M}$  checks whether  $\mathcal{A}$  guesses the random bit correctly in the Test query. If  $\mathcal{A}$  succeeds,  $\mathcal{M}$  outputs 1; otherwise,  $\mathcal{M}$  outputs 0.

Let  $\text{Event}_{\Sigma}^{\text{pw}_A}(\mathcal{M})/\text{Event}_{\Sigma}^{\text{pw}'_A}(\mathcal{M})$  denote that  $\mathcal{M}$  gets the challenge ciphertext of the password  $\text{pw}_A/\text{pw}'_A$  and outputs 1. If  $\mathcal{M}$  gets the challenge ciphertext of  $\text{pw}_A/\text{pw}'_A$ , the environment provided by  $\mathcal{M}$  for the protocol adversary  $\mathcal{A}$  is the same as experiment  $\mathcal{T}_8/\mathcal{T}_9$ . Therefore, we have  $\Pr[\text{Event}_{\Sigma}^{\text{pw}_A}(\mathcal{M}) = 1] = \Pr[\text{Success}_{\mathcal{T}_8}]$  and  $\Pr[\text{Event}_{\Sigma}^{\text{pw}'_A}(\mathcal{M}) = 1] = \Pr[\text{Success}_{\mathcal{T}_9}]$ . Let  $\text{Adv}_{\Sigma, \mathcal{M}}^{\text{IND-CCA}}(\kappa)$  be  $\mathcal{M}$ 's advantage in attacking the encryption system  $\Sigma$ , then

$$\begin{aligned} |\text{Adv}_{\mathcal{T}_9}(\kappa) - \text{Adv}_{\mathcal{T}_8}(\kappa)| &= 2|\Pr[\text{Success}_{\mathcal{T}_9}] - \Pr[\text{Success}_{\mathcal{T}_8}]| \\ &= 2|\Pr[\text{Event}_{\Sigma}^{\text{pw}'_A}(\mathcal{M}) = 1] - \Pr[\text{Event}_{\Sigma}^{\text{pw}_A}(\mathcal{M}) = 1]| \\ &= 2\text{Adv}_{\Sigma, \mathcal{M}}^{\text{IND-CCA}}(\kappa). \end{aligned} \quad (6)$$

According to the CCA security of encryption system  $\Sigma$ , the lemma holds.

So far, we have completed the modification of  $\text{Send}$  query. We now analyze the adversary's advantage in the final experiment  $\mathcal{T}_9$ . If adversary  $\mathcal{A}$  cannot guess the correct password,  $\mathcal{A}$  can only rely on guessing the random bit  $b$  in the Test query to succeed. Note that all session keys are replaced

with random tapes, so the probability of  $\mathcal{A}$  guessing  $b$  is only 1/2. At the same time, as described in experiments  $\mathcal{T}_5$  and  $\mathcal{T}_6$ , the adversary  $\mathcal{A}$  can succeed by guessing the password, and the probability of each correct guess is at most  $1/|\mathcal{D}|$ , so the ultimate advantage of adversary  $\mathcal{A}$  is at most  $q_{\text{send}}/|\mathcal{D}|$ . Combining the conclusions of Lemma 1 to Lemma 9, we can see that  $\text{Adv}_{\mathcal{T}_9}(n) \leq q_{\text{send}}/|\mathcal{D}| + \text{negl}(\kappa)$ , that is, the conclusion of Theorem 1 is established.

## 4. A Two-Round 3PAKE Protocol

In this section, we propose a two-round 3PAKE protocol based on the two-round 2PAKE protocol in Section 3. Client A and server C share the password  $\text{pw}_A$ , and client B and server C share the password  $\text{pw}_B$ . The primitives and the initialization process here are the same as the two-round 2PAKE protocol above. The clients and server implement the honest 3PAKE protocol on lattice, as shown in Table 3.

**4.1. Protocol Execution.** Clients A and B, respectively, choose random tapes  $r_{1A} \leftarrow_r \{0, 1\}^*$  and  $r_{1B} \leftarrow_r \{0, 1\}^*$  for encryption hash keys  $k_{1A}, k_{1B} \leftarrow_r K$ . Then, A/B computes the projection key  $s_{1A} = \alpha(k_{1A})/s_{1B} = \alpha(k_{1B})$  and sets  $\text{label}_{1A} = A\|B\|S\|s_{1A}/\text{label}_{1B}B\|A\|S\|s_{1B}$ . A/B continues to compute  $c_{1A} = (u_{1A}, v_{1A}) = \Sigma(\text{pk}, \text{label}_{1A}, \text{pw}_A, r_{1A})/c_{1B} = (u_{1B}, v_{1B}) = \Sigma(\text{pk}, \text{label}_{1B}, \text{pw}_B, r_{1B})$ . Finally, client A/B sends to server C a message  $(A\|B\|S\|s_{1A}\|c_{1A} = (u_{1A}, v_{1A}))/ (B\|A\|S\|s_{1B}\|c_{1B} = (u_{1B}, v_{1B}))$ .

After receiving  $(A\|B\|S\|s_{1A}\|c_{1A})$  and  $(B\|A\|S\|s_{1B}\|c_{1B})$  from clients A and B, the server C checks whether  $c_{1A}$  and  $c_{1B}$  are valid ciphertexts with respect to  $\text{pk}$ ,  $\text{label}_{1A}$  and  $\text{label}_{1B}$ . If not, C refuses and the protocol is terminated. Otherwise, C chooses hash keys  $k_{2A}, k_{2B} \leftarrow_r K$  and  $k_{2A}^*, k_{2B}^* \leftarrow_r K$ . It computes  $s_{2A} = \alpha(k_{2A}), s_{2B} = \alpha(k_{2B}), s_{2A}^* = \alpha(k_{2A}^*), s_{2B}^* = \alpha(k_{2B}^*), r_{jA}\|\tau_{jA}\|\text{sk}_{jA} \leftarrow H_{k_{2A}^*}(u_{1A}, \text{pw}_A), m_A = \tau_{jA} \oplus \text{sk}_{jB}, m_B = \tau_{jB} \oplus \text{sk}_{jA}, r_{jB}\|\tau_{jB}\|\text{sk}_{jB} \leftarrow H_{k_{2B}^*}(u_{1B}, \text{pw}_B), u_{2A} = f(\text{pk}, \text{pw}_A, r_{jA}), u_{2B} = f(\text{pk}, \text{pw}_B, r_{jB}), \text{tk}_A = \text{Hash}(s_{1A}, (u_{2A}, \text{pw}_A) r_{jA}) \oplus H_{k_{2A}}(u_{1A}, \text{pw}_A), \text{tk}_B = \text{Hash}(s_{1B}, (u_{2B}, \text{pw}_B), r_{jB}) \oplus H_{k_{2B}}(u_{1B}, \text{pw}_B), \Delta_A = \text{tk}_A \oplus \text{ECC}(H_{k_{2A}^*}(u_{1A}, \text{pw}_A)),$  and  $\Delta_B = \text{tk}_B \oplus \text{ECC}(H_{k_{2B}^*}(u_{1B}, \text{pw}_B))$ . Then, C sets  $\text{label}_{2A} = A\|B\|C\|s_{1A}\|s_{2A}\|s_{2A}^*\|\Delta_A\|c_{1A}$  and  $\text{label}_{2B} = B\|A\|C\|s_{1B}\|s_{2B}\|s_{2B}^*\|\Delta_B\|c_{1B}$  and computes  $v_{2A} = g(\text{pk}, \text{label}_{2A}, \text{pw}_A, r_{jA}), v_{2B} = g(\text{pk}, \text{label}_{2B}, \text{pw}_B, r_{jB})$ . Finally, C sends to A/B the message  $(s_{2A}\|s_{2A}^*\|\Delta_A\|c_{2A} = (u_{2A}, v_{2A})\|m_A)/(s_{2B}\|s_{2B}^*\|\Delta_B\|c_{2B} = (u_{2B}, v_{2B})\|m_B)$  and outputs  $\text{sk}_{AB} = \text{sk}_{jA} \oplus \text{sk}_{jB}$ .

Received from server C  $(s_{2A}\|s_{2A}^*\|\Delta_A\|c_{2A} = (u_{2A}, v_{2A})\|m_A)/(s_{2B}\|s_{2B}^*\|\Delta_B\|c_{2B} = (u_{2B}, v_{2B})\|m_B)$ , client A/B checks whether  $c_{2A}/c_{2B}$  is a valid ciphertext with respect to  $\text{pk}$  and  $\text{label}_{2A}/\text{label}_{2B}$ . If not, A/B rejects and the protocol aborts. Otherwise, A/B computes  $\text{tk}'_A = H_{k_{1A}}(u_{2A}, \text{pw}_A) \oplus \text{Hash}(s_{2A}, (u_{1A}, \text{pw}_A), r_{1A})/\text{tk}'_B = H_{k_{1B}}(u_{2B}, \text{pw}_B) \oplus \text{Hash}(s_{2B}, (u_{1B}, \text{pw}_B), r_{1B}), H'_A = \text{ECC}^{-1}(\text{tk}'_A \oplus \Delta_A)/H'_B = \text{ECC}^{-1}(\text{tk}'_B \oplus \Delta_B)$ . Client A/B then checks whether the Hamming distance between  $H'_A/H'_B$  and  $\text{Hash}(s_{2A}^*, (u_{1A}, \text{pw}_A), r_{1A})/\text{Hash}(s_{2B}^*, (u_{1B}, \text{pw}_B), r_{1B})$  is less than  $2\epsilon/\ell$ . If not, Client A/B rejects and the protocol aborts. Otherwise, A/B sets  $r_{iA}\|\tau_{iA}\|\text{sk}_{iA}$

TABLE 3: An honest execution of two-round 3PAKE protocol.

Client A	Two-round 3PAKE protocol Server	Client B
$r_{1A} \leftarrow_r \{0, 1\}^*$		$r_{1B} \leftarrow_r \{0, 1\}^*$
$k_{1A} \leftarrow_r K$		$k_{1B} \leftarrow_r K$
$s_{1A} = \alpha(k_{1A})$		$s_{1B} = \alpha(k_{1B})$
$\text{label}_{1A} = A \  B \  S \  s_{1A}$		$\text{label}_{1B} = B \  A \  S \  s_{1B}$
$u_{1A} = f(\text{pk}, \text{pw}_A, r_{1A})$		$u_{1B} = f(\text{pk}, \text{pw}_B, r_{1B})$
$v_{1A} = g(\text{pk}, \text{label}_{1A}, \text{pw}_A, r_{1A})$		$v_{1B} = g(\text{pk}, \text{label}_{1B}, \text{pw}_B, r_{1B})$
$\underbrace{A \  B \  S \  s_{1A} \  c_{1A} = (u_{1A}, v_{1A})}_{\leftarrow}$		$\underbrace{B \  A \  S \  s_{1B} \  c_{1B} = (u_{1B}, v_{1B})}_{\leftarrow}$
	$k_{2A} \leftarrow_r K, k_{2A}^* \leftarrow_r K$	$k_{2B} \leftarrow_r K, k_{2B}^* \leftarrow_r K$
	$s_{2A} = \alpha(k_{2A}), s_{2A}^* = \alpha(k_{2A}^*)$	$s_{2B} = \alpha(k_{2B}), s_{2B}^* = \alpha(k_{2B}^*)$
	$r_{jA} \  r_{jA} \  \text{sk}_{jA} \leftarrow H_{k_{2A}}(u_{1A}, \text{pw}_A)$	$r_{jB} \  \tau_{jB} \  \text{sk}_{jB} \leftarrow H_{k_{2B}}(u_{1B}, \text{pw}_B)$
	$m_A = \tau_{jA} \oplus \text{sk}_{jB}$	$m_B = \tau_{jB} \oplus \text{sk}_{jA}$
	$u_{2A} = f(\text{pk}, \text{pw}_A, r_{jA})$	$u_{2B} = f(\text{pk}, \text{pw}_B, r_{jB})$
$\text{tk}_A = \text{Hash}(s_{1A}, (u_{2A}, \text{pw}_A), r_{jA}) \oplus H_{k_{2A}}(u_{1A}, \text{pw}_A)$		$\text{tk}_B = \text{Hash}(s_{1B}, (u_{2B}, \text{pw}_B), r_{jB}) \oplus H_{k_{2B}}(u_{1B}, \text{pw}_B)$
$\Delta_A = \text{tk}_A \oplus \text{ECC}(H_{k_{2A}}(u_{1A}, \text{pw}_A))$		$\Delta_B = \text{tk}_B \oplus \text{ECC}(H_{k_{2B}}(u_{1B}, \text{pw}_B))$
$\text{label}_{2A} = A \  B \  C \  s_{1A} \  s_{2A} \  s_{2A}^* \  \Delta_A \  c_{1A}$		$\text{label}_{2B} = B \  A \  C \  s_{1B} \  s_{2B} \  s_{2B}^* \  \Delta_B \  c_{1B}$
$v_{2A} = g(\text{pk}, \text{label}_{2A}, \text{pw}_A, r_{jA})$		$v_{2B} = g(\text{pk}, \text{label}_{2B}, \text{pw}_B, r_{jB})$
$\underbrace{s_{2A} \  s_{2A}^* \  \Delta_A \  c_{2A} = (u_{2A}, v_{2A}) \  m_A}_{\leftarrow}$		$\underbrace{s_{2B} \  s_{2B}^* \  \Delta_B \  c_{2B} = (u_{2B}, v_{2B}) \  m_B}_{\leftarrow}$
$\text{tk}'_A = H_{k_{1A}}(u_{2A}, \text{pw}_A) \oplus \text{Hash}(s_{2A}, (u_{1A}, \text{pw}_A), r_{1A})$		$\text{tk}'_B = H_{k_{1B}}(u_{2B}, \text{pw}_B) \oplus \text{Hash}(s_{2B}, (u_{1B}, \text{pw}_B), r_{1B})$
$H'_A = \text{ECC}^{-1}(\text{tk}'_A \oplus \Delta_A)$		$H'_B = \text{ECC}^{-1}(\text{tk}'_B \oplus \Delta_B)$
If $\text{Ham}(H'_A, \text{Hash}(s_{2A}^*, (u_{1A}, \text{pw}_A), r_{1A})) \leq 2\epsilon/\ell$		If $\text{Ham}(H'_B, \text{Hash}(s_{2B}^*, (u_{1B}, \text{pw}_B), r_{1B})) \leq 2\epsilon/\ell$
$r_{iA} \  r_{iA} \  \text{sk}_{iA} \leftarrow H'_A$		$r_{iB} \  \tau_{iB} \  \text{sk}_{iB} \leftarrow H'_B$
$\text{sk}_{AB} = m_A \oplus \tau_{iA} \oplus \text{sk}_{iA}$		$\text{sk}_{BA} = m_B \oplus \tau_{iB} \oplus \text{sk}_{iB}$

$\leftarrow H'_A / r_{iB} \| \tau_{iB} \| \text{sk}_{iB} \leftarrow H'_B$  and outputs  $\text{sk}_{AB} = m_A \oplus \tau_{iA} \oplus \text{sk}_{iA} / \text{sk}_{BA} = m_B \oplus \tau_{iB} \oplus \text{sk}_{iB}$ .

**4.2. Correctness.** After the protocol is executed honestly, the probability of a mismatch between the session keys obtained by the two clients A and B is negligible. From the approximate correctness of the NA-ASPH, the probability that the Hamming distance between  $\text{tk}_A$  ( $\text{tk}_B$ ) calculated by client A (B) and  $\text{tk}'_A$  ( $\text{tk}'_B$ ) calculated by the clients is greater than  $(\epsilon(n) \cdot n)$  can be neglected. Then, from the definition of error correction code ECC, client A (B) and server C can obtain the same  $r_{iA} \| \tau_{iA} \| \text{sk}_{iA}$  ( $r_{jB} \| \tau_{jB} \| \text{sk}_{jB}$ ), so A and B can get the same session key,

$$\begin{aligned} \text{sk}_{AB} &= m_A \oplus \tau_{iA} \oplus \text{sk}_{iA} = \tau_{jA} \oplus \text{sk}_{jB} \oplus \tau_{iA} \oplus \text{sk}_{iA} \\ &= \text{sk}_{jB} \oplus \text{sk}_{iA} = \tau_{jB} \oplus \text{sk}_{jA} \oplus \tau_{iB} \oplus \text{sk}_{iB} \\ &= m_B \oplus \tau_{iB} \oplus \text{sk}_{iB} = \text{sk}_{AB}. \end{aligned} \quad (7)$$

**4.3. Security.** Since the protocol here is symmetrical with respect to the clients, the proof usually can only take one client as an example. The security proof of the protocol follows Section 3 closely. We outline the main ideas. First, based on the CCA security of the underlying primitive  $\Sigma$ , the adversary cannot obtain any useful information about the real password

through Execute query. In the Execute query, if the simulator replaces the valid password with an illegal one, guaranteed by the smoothness of NA-ASPH, the adversary cannot distinguish the corresponding two experiments computationally. Second, if the adversary simply replays the messages between participants, the proof is the same as the Execute query. Third, if the adversary modifies the output message of some instances (that is, it modifies (label, c)), the simulator can obtain the corresponding plaintext  $\text{pw}_{A/B}^{\text{ad}}$  through the decryption oracle provided by CCA security. If  $\text{pw}_{A/B}^{\text{ad}} = \text{pw}_{A/B}$  holds, the corresponding attack is successful, which will increase the adversary's advantage. Using the CCA security of  $\Sigma$ ,  $\text{pw}_{A/B}$  is uniformly sampled from the password dictionary  $\mathcal{D}$ , so  $\Pr[\text{pw}_{A/B}^{\text{ad}} = \text{pw}_{A/B}] \leq 1/|\mathcal{D}|$ . Assuming that the adversary can perform at most  $Q(\kappa)$  online attacks, then the adversary's advantage is at most  $Q(\kappa)/|\mathcal{D}|$ . And if  $\text{pw}_{A/B}^{\text{ad}} \neq \text{pw}_{A/B}$  holds, then from the adversary's view, the session key obtained is indistinguishable from the uniform distribution (according to the smoothness of NA-ASPH).

## 5. Protocol Performance Analysis

In this section, we will compare the performance of the two proposed protocols with other related protocols in terms of safety and efficiency. The comparison results are shown in Table 4, where Type represents the protocol type, M-Auth

TABLE 4: Performance comparisons of PAKE protocols.

Protocol	Type	M-Auth	Anti-Qu	Round	C-method	C-cost
Z-2PAKE [1]	2-party	✓	✓	2	V-mul	$(2m+2n_1)lbq+n$
K-2PAKE [33]	2-party	✗	✓	3	V-mul	$(mn+2n)lbq+3n$
D-2PAKE [34]	2-party	✓	✓	3	V-mul	$(mn/2+m/2+3n)lbq+2n$
2PAKE	2-party	✓	✓	2	V-mul	$(2m+3n_1)lbq+n$
A-3PAKE [30]	3-party	✗	✗	4	Exp	—
Y-3PAKE [35]	3-party	✓	✓	3	V-mul	$2[(mn+n)lbq+3n]$
X-3PAKE [36]	3-party	✓	✓	3	V-mul	$7nlbq+9n+5$
3PAKE	3-party	✓	✓	2	V-mul	$2[(2m+3n_1)lbq+2n]$

indicates whether the protocol can provide mutual authentication, Round denotes the number of communication rounds required by the protocol, Anti-Qu represents whether the protocol can resist quantum attacks, C-method indicates the operation method of the protocol, C-cost represents the communication cost of the protocol, V-mul denotes vector multiplication, Exp indicates exponentiation, and  $n = n_1 + n_2$ .

In terms of security, we mainly compare with other protocols in (1) whether it can resist quantum attacks; (2) whether it can achieve mutual authentication. In terms of efficiency, we mainly compare from the following three aspects: (1) the selection of cryptographic primitives, (2) the calculation method, and (3) the communication overhead. Note that the calculation method adopted is used to roughly measure the computational cost of the corresponding protocol. Moreover, the computational cost of modular exponential operations is much greater than linear operations on matrices and vectors.

Compared with the K-PAKE [33] and D-PAKE [34], the advantage of the 2PAKE is that mutual authentication and key exchange can be achieved within two rounds of transmission. And the size of the ciphertext of K-PAKE and D-PAKE are  $O(n)$  larger than 2PAKE. The size of the projection key of K-PAKE and D-PAKE is determined by the ciphertext and the hash key, larger than that of 2PAKE.

Compared with the typical three-party PAKE, A-3PAKE [30], the 3PAKE in this paper is lattice-based and can resist quantum attacks. The proposed 3PAKE can achieve mutual authentication within two rounds of transmission. In addition, A-3PAKE uses exponential operation, while the 3PAKE protocol uses vector multiplication, which has higher computational efficiency.

Compared with Y-3PAKE [35] and X-3PAKE [36] protocols, 3PAKE only requires 2 rounds of transmission. The communication cost of Y-3PAKE protocol mainly depends on the size of the ciphertext, the projection key, and the message authentication code. The size of the ciphertext is  $O(n)$  larger than 3PAKE. The size of the projection key of Y-3PAKE is determined by the ciphertext and the hash key, larger than that of 3PAKE. In addition, the Y-PAKE protocol needs to calculate and send a message authentication code for mutual authentication, while 3PAKE performs mutual authentication by verifying the validity of the ciphertext. The amount of messages that needs to be transmitted in X-PAKE is large, resulting in increased communication overhead. Therefore, the

communication overhead of the Y-3PAKE and X-PAKE protocols is greater than 3PAKE in this paper.

Z-PAKE [1] is also a two-round protocol, but it is designed for two parties. Compared with the Z-PAKE protocol, 2PAKE adds a projection key to the communication overhead. However, if the three-party PAKE based on Z-PAKE is implemented in a traditional way, at least 4 rounds of communication, that is, 8 message transmissions, are required. But the 3PAKE in this paper only needs 2 rounds of communication, namely 4 message transmissions.

The protocols in this paper have advantages of efficiency and security over traditional protocols based on finite fields, since lattice operations (vector multiplication) are more efficient than exponentiation and lattice problems remain hard for quantum attacks and subexponential-time adversaries. Both protocols in this article can achieve mutual authentication; thus, they can resist imperceptible on-line dictionary attacks. Besides, they are both two-round protocols with less number of transmissions. And the underlying primitive is an improved lattice-based CCA-secure PKE, which can reduce encryption parameters and further reduce computational overhead. In particular, compared with other three-party protocols, the two-round 3PAKE protocol proposed has smaller communication and computation overhead, so it is adaptable to large-scale communication systems.

## 6. Conclusions

This paper proposes two password-based authenticated key exchange protocols based on the LWE problem from lattices, which can resist quantum attacks and have high efficiency. In the random oracle model, this paper gives a strict security proof of the proposed protocols. In addition, the proposed PAKE protocols can achieve mutual authentication in two rounds of transmission. And the 3PAKE protocol is practical for large-scale communication systems. Compared with the existing related protocols, the protocols in this paper have higher security and lower communication and computing overhead. In our protocols, the client's password is stored on a single server, so the proposed protocols are not resistant to hacker attacks. In the future, we will study the multiserver PAKE protocol that can resist hacker attacks.

## Data Availability

The extra data used to support the findings of this study are available from the corresponding author. Email: guo\_yuanbo@126.com.

## Conflicts of Interest

The authors declare no competing financial interest.

## Acknowledgments

This work has been supported by the National Natural Science Foundation of China (Grant No. 61501515) and Foundation of Science and Technology on Information Assurance Laboratory (No. KJ-15-108).

## References

- [1] J. Zhang and Y. Yu, "Two-Round PAKE from Approximate SPH and Instantiations from Lattices," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 37–67, Hong Kong, China, 2017.
- [2] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.
- [3] J. Zhao and D. Gu, "Provably secure three-party password-based authenticated key exchange protocol," *Information Sciences*, vol. 184, no. 1, pp. 310–323, 2012.
- [4] M. S. Farash, S. H. Islam, and M. S. Obaidat, "A provably secure and efficient two-party password-based explicit authenticated key exchange protocol resistance to password guessing attacks," *Concurrency & Computation Practice & Experience*, vol. 27, no. 17, pp. 4897–4913, 2017.
- [5] A. Groce and J. Katz, "A new framework for efficient password-based authenticated key exchange," in *Proceedings of the ACM Conference on Computer and Communications*, pp. 516–525, Chicago, Illinois, 2010.
- [6] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.
- [7] J. Katz, R. Ostrovsky, and M. Yung, "Efficient and secure authenticated key exchange using weak passwords," *Journal of the Association for Computing Machinery*, vol. 57, no. 1, pp. 79–117, 2010.
- [8] Z. Li and D. Wang, "Achieving one-round password-based authenticated key exchange over lattices," *IEEE Transactions on Services Computing*, vol. 2019, no. 8, pp. 1–14, 2019.
- [9] R. Gennaro and Y. Lindell, "A framework for password-based authenticated key exchange<sup>1</sup>," *ACM Transactions on Information & System Security*, vol. 9, no. 2, pp. 181–234, 2006.
- [10] S. Bellare and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks," in *2012 IEEE Symposium on Security and Privacy*, p. 72, Oakland, California, 1992.
- [11] T. Lomas, L. Gong, J. Saltzer, and R. Needham, "Reducing risks from poorly chosen keys," in *Proceedings of the twelfth ACM symposium on Operating systems principle (SOSP '89)*, pp. 14–18, New York, NY, 1989.
- [12] L. Gong, M. A. Lomas, R. M. Needham, and J. H. Saltzer, "Protecting poorly chosen secrets from guessing attacks," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, pp. 648–656, 1993.
- [13] S. M. Bellare and M. Merritt, "Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password file compromise," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 244–250, New York, NY, 1989.
- [14] L. Gong, "Optimal authentication protocols resistant to password guessing attacks," in *Proceedings of the 8th IEEE Computer Security Foundations Workshop*, pp. 24–29, Los Alamitos, California, 1995.
- [15] M. Steiner, G. Tsudik, and M. Waidner, "Refinement and extension of encrypted key exchange," *Acm Sigops Operating Systems Review*, vol. 29, no. 3, pp. 22–30, 1995.
- [16] W. Thomas, "The Secure Remote Password Protocol," in *Proceedings of the Network and Distributed System Security Symposium (NDSS'98)*, pp. 97–111, San Diego, California, 2000.
- [17] M. B. Wllare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proceedings of the Advances in Cryptology (Eurocrypt'00)*, pp. 139–155, Berlin, Germany, 2000.
- [18] V. Boyko, P. D. Mackenzie, and S. Patel, "Provably secure password-authenticated key exchange using Diffie-Hellman," in *Proceedings of Advances in Cryptology (Eurocrypt'00)*, pp. 156–171, Berlin, Germany, 2000.
- [19] O. A. Goldreich and Y. Lindell, "Session-Key generation using human passwords only," *Journal of Cryptology*, vol. 19, no. 3, pp. 241–340, 2006.
- [20] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *EUROCRYPT 2000: International Conference on the Theory and Application of Cryptographic Techniques*, pp. 139–155, Bruges, Belgium, 2000.
- [21] V. Boyko, P. Mac Kenzie, and S. Patel, "Provably secure password-authenticated key exchange using Diffie-Hellman," in *EUROCRYPT 2000: International Conference on the Theory and Application of Cryptographic Techniques*, pp. 156–171, Bruges, Belgium, 2000.
- [22] E. Bresson, O. Chevassut, and D. Pointcheval, "Security proofs for an efficient password-based key exchange," in *Proceedings of the 10th ACM conference on Computer and communications security*, pp. 241–250, Washington DC, 2003.
- [23] M. Abdalla, F. Benhamouda, and D. Pointcheval, "Disjunctions for hash proof systems: new constructions and applications," in *EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 69–100, Sofia, Bulgaria, 2015.
- [24] F. Benhamouda, O. Blazy, C. Chevalier, D. Pointcheval, and D. Vergnaud, "New techniques for SPHF's and efficient one-round PAKE protocols," in *Advances in Cryptology-CRYPTO 2013*, pp. 449–475, Santa Barbara, CA, 2013.
- [25] J. Katz and V. Vaikuntanathan, "Round-optimal password-based authenticated key exchange," in *8th Theory of Cryptography Conference*, pp. 293–310, Providence, RI, 2011.
- [26] S. Jiang and G. Gong, "Password Based Key Exchange with Mutual Authentication," in *Selected Areas in Cryptography (SAC 2004)*, pp. 267–279, Waterloo, Canada, 2004.
- [27] Y. Mao, *Research on Password-Based Authenticated Key Exchange Protocols and Associated Encryption Algorithms from Lattices*, PLA Information Engineering University, 2012.

- [28] J. Katz, P. MacKenzie, G. Taban, and V. Gligor, "Two-server password-only authenticated key exchange," *Journal of Computer & System Sciences*, vol. 78, no. 2, pp. 651–669, 2005.
- [29] R. Mario and G. Raimondo, "Provably secure threshold password-authenticated key exchange," *Journal of Computer and System Sciences*, vol. 72, no. 6, pp. 507–523, 2006.
- [30] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-Based Authenticated Key Exchange in the Three-Party Setting," in *8th International Workshop on Theory and Practice in Public Key Cryptography*, pp. 65–84, Les Diablerets, Switzerland, 2005.
- [31] W. Minghui and W. Jiandong, "Three-party authentication key exchange protocol based on password," *Computer Engineering*, vol. 38, no. 2, pp. 146–150, 2012.
- [32] W. Guocai, K. Fusong, and W. Fang, "ECDSA-based password authenticated key exchange protocol for threeparty," *Computer Engineering*, vol. 38, no. 6, pp. 153–155, 2012.
- [33] K. Jonathan and V. Vaikuntanathan, "Smooth Projective Hashing and Password-Based Authenticated Key Exchange from Lattices," in *Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, pp. 636–652, Tokyo, Japan, 2009.
- [34] Y. Ding and L. Fan, "Efficient password-based authenticated key exchange from lattices," *International Journal of Advancements in Computing Technology*, vol. 1, no. 22, pp. 934–938, 2011.
- [35] M. Ye, X. Hu, and W. Liu, "Password authenticated key exchange protocol in the three party setting based on lattices," *Journal of Electronics & Information Technology*, vol. 35, no. 6, pp. 1376–1381, 2013.
- [36] D. Xu, "Provably Secure Three-party Password Authenticated Key Exchange Protocol Based on Ring Learning with Error," *IACR Cryptol. ePrint Arch*, vol. 2017, p. 360, 2017.
- [37] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in *EUROCRYPT 2002: International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 45–64, Amsterdam, The Netherlands, 2002.



## Research Article

# Privacy-Preserving Graph Operations for Mobile Authentication

Peng Li <sup>1,2</sup>, Fucai Zhou <sup>1</sup>, Zifeng Xu <sup>1</sup>, Yuxi Li <sup>1</sup>, and Jian Xu <sup>1</sup>

<sup>1</sup>Software College, Northeastern University, Shenyang, China

<sup>2</sup>School of Information Engineering, Eastern Liaoning University, China

Correspondence should be addressed to Fucai Zhou; [fczhou@mail.neu.edu.cn](mailto:fczhou@mail.neu.edu.cn)

Received 7 July 2020; Revised 13 October 2020; Accepted 11 November 2020; Published 23 November 2020

Academic Editor: Ding Wang

Copyright © 2020 Peng Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Along with the fast development of wireless technologies, smart devices have become an integral part of our daily life. Authentication is one of the most common and effective methods for these smart devices to prevent unauthorized access. Moreover, smart devices tend to have limited computing power, and they may possess sensitive data. In this paper, we investigate performing graph operations in a privacy-preserving manner, which can be used for anonymous authentication for smart devices. We propose two protocols that allow two parties to jointly compute the intersection and union of their private graphs. Our protocols utilize homomorphic encryption to prevent information leakage during the process, and we provide security proofs of the protocols in the semihonest setting. At last, we implement and evaluate the efficiency of our protocols through experiments on real-world graph data.

## 1. Introduction

With the rapid development of IoT technology, we are surrounded by various types of smart devices in our daily life, such as sensors, wearable devices, and smart vehicles [1]. Authentication is one of the most important mechanisms to provide security protection for these smart devices [2], and authentication for light-weighted devices has become a hot research topic in the past years [3, 4].

In recent years, researchers have proposed several mobile authentication schemes based on graph data structure and graph algorithms [5–7]. Graph data and graph processing are well studied for the last decades [8, 9], since they can help to solve many practical problems in different application areas, such as web data processing [10], data mining [11], social networking [12], biological networking [13], and communication networking [14].

*1.1. Motivation.* In this paper, we consider the problem of computing graph operations between two parties while preventing information leakage, which has great potential in smart device authentication. For example, when the mobile devices communicate with cloud servers, they need to first jointly perform identity authentication for security protec-

tion. Since the mobile devices may contain sensitive information of the users and the cloud servers cannot be fully trusted in general, the privacy leakage problem for mobile authentication has become a security threat [15]. In order to protect the privacies of the mobile devices, the devices can model their identities and properties as graph-structured data, and the cloud servers can model their authentication policies as graph-structured data as well. After that, the identity authentication process can be converted into performing graph operations in a privacy-preserving manner.

*1.2. Our Contributions.* We study the problem of performing graph intersection and union while protecting the privacies of the input graphs. Suppose that for two parties, Alice and Bob, each has a private graph, denoted as  $G_A$  and  $G_B$ , respectively. Alice wishes to learn the intersection and union of these two graphs. In other words, Alice wishes to learn  $G_I = G_A \cap G_B$  and  $G_U = G_A \cup G_B$ . In addition, both Alice and Bob do not wish to reveal any information about their graphs to the other party. The contributions of this paper can be summarized as below:

- (i) We present two graph operation protocols between two parties, a server and a client. The first protocol

allows the server and the client to jointly compute the intersection of their input graphs, and the second protocol computes the union of the input graphs. Our constructions first use the Paillier cryptosystem and oblivious polynomial evaluation to compute the intersection and the union of the vertices. After that, we use the homomorphic property of the Paillier cryptosystem to compute the edge intersection and union

- (ii) We provide the security models of the protocols, and we prove that the protocols are secure in the semi-honest setting. Furthermore, we analyze the information leakage and propose methods to minimize the leakages
- (iii) We discuss the efficiencies of the protocols in terms of computation costs and communication costs. At last, we implement our constructions and perform experiments on real-world graph data

An earlier version of this paper was presented at the 22nd Australasian Conference on Information Security and Privacy, 2017 [16]. The previous work presented a private graph intersection protocol with rough analysis. This paper extends the previous work by presenting a private graph union protocol with detailed analysis and experimental results.

## 2. Related Work

There are many different approaches to construct authentication schemes for smart devices. Among them, graph-based authentication schemes are widely used in IoT [5, 17, 18]. In 2002, Micali and Rivest [19] first introduced the transitive signature based on graph theory, which provides an unforgeable signature for undirected graphs. After that, various graph-based signature and authentication schemes were proposed [5–7]. In 2017, Chuang et al. [5] proposed an authentication system in Internet of Things based on multigraph zero-knowledge. The system provides suitable security protection for IoT authentication services. The proposed multigraph zero-knowledge procedure is faster than traditional zero-knowledge methods and ECC-based solutions. The experiment results indicate that the system is lightweighted and highly adaptive. Lin et al. [6] proposed a transitively graph authentication scheme for blockchain-based identity management systems in 2018. The system is used to bind a digital identity object to its real-world entity, therefore achieving identity authentication. The system is constructed based on transitively closed undirected graphs and vertex signatures. According to the evaluation results, the system is efficient, even when the graph dynamically adds or deletes vertices and edges. In 2019, Shao et al. [7] proposed a multifactor authentication scheme using a fuzzy graph domination model. The scheme is adaptive choosing one or multiple privacy-preserving identities to authenticate the users. The authors designed a weighted vertex-edge dominating set to solve the weighted domination problem on fuzzy graphs. Compared to existing solutions, the scheme is more efficient for solving instances with moderate orders.

In this work, we consider the problem of performing graph intersection and union in the privacy-preserving manner and proposed two secure multiparty computation protocols. Secure Multiparty computation (MPC) has been extensively studied over the past decades. Generally speaking, MPC allows multiple participants to jointly perform certain computations without losing the privacy of their input data, even when some players cheat during the process. MPC was first formally introduced by Yao in 1982 [20] and extended by Goldreich et al. [21]. Their works convert certain computation problems into a combinatorial circuit, then the parties perform computations over the gates in the circuit. After that, a large number of MPC protocols have been proposed to solve various problems, such as privacy-preserving set operations [22] and private information retrieval [23].

## 3. Preliminary

In this section, we present the preliminaries related to our proposed protocols. First, we present the relevant notations that we used in this paper in Table 1.

**3.1. Additive Homomorphic Encryption.** Homomorphic encryption schemes allow the users to perform certain computation operations on the ciphertext space, such as addition and multiplication. In our private graph operation protocols, we utilize an additive homomorphic encryption scheme called the Paillier cryptosystem, proposed by Paillier in 1999 [24]. The Paillier cryptosystem contains three algorithms, described as follows:

$(pk, sk) \leftarrow \text{KeyGen}(1^k)$  is the key generation algorithm. The input is a security parameter  $k$ . The outputs are a public key  $pk$  and a secret key  $sk$ . The public key contains a large number  $N$  which specifies the message space, the ciphertext space, and the random space to be  $\mathbb{Z}_N$ ,  $\mathbb{Z}_{N^2}^*$ , and  $\mathbb{Z}_N^*$ , respectively.

$t^\oplus \leftarrow \text{Enc}(pk, t; r)$  is the encryption algorithm. The input is the public key  $pk$ , a plaintext  $t \in \mathbb{Z}_N$ , and a random number  $r \in \mathbb{Z}_N^*$ . The output is the ciphertext  $t^\oplus \in \mathbb{Z}_{N^2}^*$ . For simplicity, we use the notion  $t^\oplus = \text{Enc}(t)$ .

$t \leftarrow \text{Dec}(sk, t^\oplus)$  is the decryption algorithm. The input is the secret key  $sk$  and a ciphertext  $t^\oplus \in \mathbb{Z}_{N^2}^*$ . The output is the plaintext  $t \in \mathbb{Z}_N$ . For simplicity, we use the notion  $t = \text{Dec}(t^\oplus)$ .

The Paillier cryptosystem has the following properties:

**3.1.1. Correctness.** For any key pairs  $(pk, sk) \leftarrow \text{KeyGen}(1^k)$  and any plaintext  $t \in \mathbb{Z}_N$ ,  $\text{Dec}(\text{Enc}(t)) = t$  always holds.

**3.1.2. IND-CPA Security.** Two ciphertexts  $t_0^\oplus$  and  $t_1^\oplus$  are indistinguishable for probabilistic polynomial-time adversaries that only have access to the public parameters.

**3.1.3. Homomorphic Property.** For any two plaintexts  $t_0, t_1 \in \mathbb{Z}_N$ , there exists an operation  $\oplus$  in the ciphertext space, such that  $\text{Dec}(\text{Enc}(t_0) \oplus \text{Enc}(t_1)) = t_0 + t_1$ . Furthermore, there exists another operation  $\otimes$  in the ciphertext space, such that  $\text{Dec}(\text{Enc}(t_0) \otimes \text{Enc}(t_1)) = t_0 \cdot t_1$ .

TABLE 1: Table of notations.

Symbol	Description
PGI	Private graph intersection protocol
PGU	Private graph union protocol
$S, C$	The server, the client
$G_S, G_C$	The server's graph, the client's graph
$G_I$	The intersection of $G_S$ and $G_C$
$G_U$	The union of $G_S$ and $G_C$
$V_S, V_C, V_I, V_U$	The vertices of $G_S, G_C, G_I$ , and $G_U$
$E_S, E_C, E_I, E_U$	The edges of $G_S, G_C, G_I$ , and $G_U$
$m, n, p, q$	The number of vertices in $G_S, G_C, G_I$ , and $G_U$

**3.2. Private Set Intersection.** Private Set Intersection (PSI) is a cryptographic protocol that allows two parties, each holding a private set, to jointly compute the intersection of their sets without leaking any additional information. The first secure two-party private set intersection protocol is introduced by Freedman, Nissim, and Pinkas (FNP) in 2004 [25]. The protocol utilizes homomorphic encryption and oblivious polynomial evaluation to ensure each party learns no information about the other party's private input during the computation. Later, several other protocols have been proposed with different features and security levels [26–28].

**3.3. Graph Representation.** In our protocol, we represent a graph as  $G = (V, E)$ , where  $V$  is the vertex collection and  $E$  is the edge collection. We represent the vertex collection as a sorted set with ascending order,  $V = \{v_1, v_2, \dots, v_z\}$ , where  $z$  is the number of vertices in  $G$ ,  $v_i \in \mathbb{Z}$ , and  $v_i < v_{i+1}$  for  $1 \leq i \leq z - 1$ . We represent the edge collection as an adjacency matrix,

$$E = \begin{pmatrix} e_{1,1} & \cdots & e_{1,z} \\ \vdots & \ddots & \vdots \\ e_{z,1} & \cdots & e_{z,z} \end{pmatrix}, \quad (1)$$

where  $e_{i,j}$  is the adjacency relation between the vertices  $v_i$  and  $v_j$ , and  $e_{i,j} \in \{0, 1\}$ . If vertices  $v_i$  and  $v_j$  are adjacent, i.e., there is at least one edge that connects them,  $e_{i,j} = 1$ ; otherwise,  $e_{i,j} = 0$ . Note that  $E$  is a square matrix with  $z$  rows and  $z$  columns. For an undirected graph,  $E$  is a symmetric matrix, since the edges are two-way.

For example, we represent the directed graph illustrated in Figure 1 as  $G = (V, E)$ , where  $V = \{1, 5, 23, 50, 74\}$  and

$$E = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \quad (2)$$

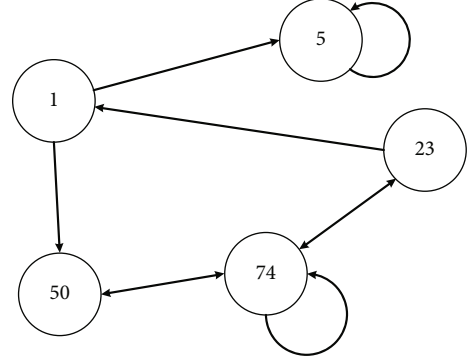


FIGURE 1: Example graph.

## 4. Definitions and Security Models

**4.1. Protocol Definitions.** We formally describe the private graph intersection (PGI) protocol and the private graph union (PGU) protocol. The protocols involve two participants, a server and a client, denoted as  $S$  and  $C$ , respectively. Each of the participants holds a private graph, which is intended to be kept secret from the other participant.

We denote the graphs of the server and client as  $G_S = (V_S, E_S)$  and  $G_C = (V_C, E_C)$ , respectively, where  $V$  and  $E$  are the sets of vertices and edges of the graphs. The intersection of  $G_S$  and  $G_C$  is defined as  $G_I = G_S \cap G_C = (V_I, E_I)$ , where  $V_I = V_S \cap V_C$  and  $E_I = E_S \cap E_C$ . The union of  $G_S$  and  $G_C$  is defined as  $G_U = G_S \cup G_C = (V_U, E_U)$ , where  $V_U = V_S \cup V_C$  and  $E_U = E_S \cup E_C$ .

PGI and PGU allow the participants to jointly compute  $G_I$  and  $G_U$ , respectively, in a privacy-preserving manner. At the end of the protocols, only the server learns the result. The formal definitions of PGI and PGU are described as follows:

**Definition 1** (private graph intersection protocol). If both participants are honest, for any  $G_S = (V_S, E_S)$  and any  $G_C = (V_C, E_C)$ , the private graph intersection protocol computes  $G_I = G_S \cap G_C$ . At the end of the protocol, only  $S$  learns  $G_I$ .

**Definition 2** (private graph union protocol). If both participants are honest, for any  $G_S = (V_S, E_S)$  and any  $G_C = (V_C, E_C)$ , the private graph union protocol computes  $G_U = G_S \cup G_C$ . At the end of the protocol, only  $S$  learns  $G_U$ .

**4.2. Security Models.** When considering privacy protecting in authentication, the term privacy may have different definitions and properties, such as user identity and untraceability [29, 30]. In this work, the privacies of the server and the client refer to any information about their graphs. Therefore, any information about the vertices and edges of the graphs is considered as private, such as the number of vertices, the number of edges, the values of the vertices, and whether two vertices are connected by an edge.

The security goals of both PGI and PGU protocols are protecting the privacies of both the server and the client during the computation. In other words, both the server and the

client should learn no information about the graph of the other party.

We use the semihonest security model for both PGI and PGU, which means both the server and the client perform the protocols faithfully, but they may try to learn any information about the graph of the other participant. The security models are adopted from the work of [31–33].

While achieving no information leakage is the ideal goal, our protocols leak partial information during the process. The information leakages for PGI are defined as leakage functions  $\mathcal{L}_1$  and  $\mathcal{L}_2$ , and the information leakages for PGU are defined as  $\mathcal{L}_3$  and  $\mathcal{L}_4$ . The detailed information about the leakage functions are as follows:  $\mathcal{L}_1$  is the number of vertices in  $G_C$ ,  $\mathcal{L}_2$  is the vertex intersection  $V_I$  and the number of vertices in  $G_S$ ,  $\mathcal{L}_3$  is the number of vertices in  $G_C$  and the number of common vertices between  $G_S$  and  $G_C$ , and  $\mathcal{L}_4$  is the vertex union  $V_U$  and the number of vertices in  $G_S$ .

The formal definitions of security models are described as follows:

*Definition 3* (PGI security). A semihonest server learns nothing about the client's graph, beyond what can be deduced from  $G_I$  and the leakage function  $\mathcal{L}_1$ , and a semihonest client learns nothing about the server's graph, beyond the leakage function  $\mathcal{L}_2$ .

*Definition 4* (PGU security). A semihonest server learns nothing about the client's graph, beyond what can be deduced from  $G_U$  and the leakage function  $\mathcal{L}_3$ , and a semihonest client learns nothing about the server's graph, beyond the leakage function  $\mathcal{L}_4$ .

## 5. Protocol Construction

In this section, we propose the constructions of PGI and PGU. The graphs of the server and the client are represented as  $G_S = (V_S, E_S)$  and  $G_C = (V_C, E_C)$ , respectively, where  $V_S = \{s_1, s_2, \dots, s_m\}$ ,  $V_C = \{c_1, c_2, \dots, c_n\}$ ,

$$E_S = \begin{pmatrix} s_{1,1} & \cdots & s_{1,m} \\ \vdots & \ddots & \vdots \\ s_{m,1} & \cdots & s_{m,m} \end{pmatrix}, E_C = \begin{pmatrix} c_{1,1} & \cdots & c_{1,n} \\ \vdots & \ddots & \vdots \\ c_{n,1} & \cdots & c_{n,n} \end{pmatrix}. \quad (3)$$

*5.1. PGI Construction.* We use the FNP protocol [25] as a building block for computing the vertex intersection. The private graph intersection protocol is described below:

*Input:*  $S$  and  $C$  hold the graphs  $G_S = (V_S, E_S)$  and  $G_C = (V_C, E_C)$ , respectively.

*Output:*  $S$  learns  $G_I = (V_I, E_I)$ .

*Protocol:*

*Step 1.*  $S$  runs the key generation algorithm of the Paillier cryptosystem,  $(pk, sk) \leftarrow \text{KeyGen}(1^k)$ , and obtains the public key and the secret key. Then,  $S$  publishes  $pk$ .

*Step 2.*

- (a)  $S$  constructs a polynomial  $P(x) = (x - s_1)(x - s_2) \cdots (x - s_m) = \sum_{u=0}^m \alpha_u x^u$ , such that all the roots of  $P(x)$  are exactly the elements in  $V_S$ . In other words,  $P(x) = 0$  if and only if  $x \in V_S$
- (b)  $S$  encrypts each  $\alpha_i$ , for  $0 \leq i \leq m$ , under the Paillier cryptosystem, and sends the set of ciphertexts  $\alpha^\oplus = \{\alpha_i^\oplus\}_{0 \leq i \leq m}$  to  $C$

*Step 3.*

- (a) By using the homomorphic properties of the Paillier cryptosystem,  $C$  evaluates the polynomial  $P$  using each element in  $V_C$  as input. In other words,  $C$  computes  $\text{Enc}(P(c_i))$ , for  $1 \leq i \leq n$
- (b) For each polynomial evaluation,  $C$  chooses a random value  $r$  and computes  $\beta_i^\oplus = \text{Enc}(rP(c_i) + c_i)$ . Then,  $C$  sends  $\beta^\oplus = \{\beta_i^\oplus\}_{1 \leq i \leq n}$  to  $S$

*Step 4.*  $S$  decrypts all the ciphertexts received and compares the decrypted values with his vertex set  $V_S$ . If a decrypted value  $\beta_i = \text{Dec}(\beta_i^\oplus)$  has a corresponding element in  $V_S$ , it is an element of the intersection of  $V_S$  and  $V_C$ . In other words, if  $\beta_i \in V_S$ ,  $\beta_i \in V_I$ . After decrypting all the received ciphertexts, the server obtains  $V_I$ .

*Step 5.*

- (a)  $S$  uses  $V_I$  to construct an adjacency matrix  $A$  of size  $p \times p$ , where  $p$  is the number of the vertex in  $V_I$ :

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,p} \\ \vdots & \ddots & \vdots \\ a_{p,1} & \cdots & a_{p,p} \end{pmatrix}. \quad (4)$$

$A$  has the property that, for each vertex pair  $v_x \in V_I$  and  $v_y \in V_I$ , if an edge exists in  $G_S$  between vertices  $v_x$  and  $v_y$ ,  $a_{x,y} = 1$ ; otherwise,  $a_{x,y} = 0$ .

- (b)  $S$  encrypts each element in  $A$  under the Paillier cryptosystem and obtains an encrypted matrix  $A^\oplus = \text{Enc}(A)$
- (c)  $S$  sends  $A^\oplus$  and  $V_I$  to  $C$

*Step 6.*

- (a) By using  $V_I$ ,  $C$  constructs an adjacency matrix  $B$  using the same method in the last step:

$$B = \begin{pmatrix} b_{1,1} & \cdots & b_{1,p} \\ \vdots & \ddots & \vdots \\ b_{p,1} & \cdots & b_{p,p} \end{pmatrix} \quad (5)$$

(b) C computes

$$\begin{aligned} E_I^\oplus &= A^\oplus \otimes B = \begin{pmatrix} a_{1,1}^\oplus & \cdots & a_{1,p}^\oplus \\ \vdots & \ddots & \vdots \\ a_{p,1}^\oplus & \cdots & a_{p,p}^\oplus \end{pmatrix} \otimes \begin{pmatrix} b_{1,1} & \cdots & b_{1,p} \\ \vdots & \ddots & \vdots \\ b_{p,1} & \cdots & b_{p,p} \end{pmatrix} \\ &= \begin{pmatrix} a_{1,1}^\oplus \otimes b_{1,1} & \cdots & a_{1,p}^\oplus \otimes b_{1,p} \\ \vdots & \ddots & \vdots \\ a_{p,1}^\oplus \otimes b_{p,1} & \cdots & a_{p,p}^\oplus \otimes b_{p,p} \end{pmatrix} \end{aligned} \quad (6)$$

(c) C sends  $E_I^\oplus$  to S

*Step 7.* S decrypts each element in  $E_I^\oplus$  and obtains  $E_I = \text{Dec}(E_I^\oplus)$ . At last, S obtains  $G_I = (V_I, E_I)$ .

**5.2. PGU Construction.** The private graph union protocol is described below:

*Input:* S and C hold the graphs  $G_S = (V_S, E_S)$  and  $G_C = (V_C, E_C)$ , respectively.

*Output:* S learns  $G_U = (V_U, E_U)$ .

*Protocol:*

*Step 1.* Same as Step 1 of PGI.

*Step 2.* Same as Step 2 of PGI.

*Step 3.*

(a) By using the homomorphic properties of the Paillier cryptosystem, C evaluates the polynomial  $P$  using each element in  $V_C$  as input. In other words, C computes  $\text{Enc}(P(c_i))$ , for  $1 \leq i \leq n$

(b) For each polynomial evaluation, C choose a random value  $r$  and computes  $\beta_i^\oplus = \text{Enc}(P(c_i)) \otimes r$ . Then, C sends the set of all resulting ciphertexts  $\beta^\oplus = \{\beta_i^\oplus\}_{1 \leq i \leq n}$  to S

*Step 4.* S decrypts each ciphertext received as  $\beta_i = \text{Dec}(\beta_i^\oplus)$  and checks the decrypted value. If  $\beta_i = 0$ , S computes  $\gamma_i^\oplus = \text{Enc}(0)$ ; otherwise, S computes  $\gamma_i^\oplus = \text{Enc}(1)$ . Then, S sends  $\gamma^\oplus = \{\gamma_i^\oplus\}_{1 \leq i \leq n}$  to C.

*Step 5.* After receiving  $\gamma^\oplus$ , C computes  $\delta_i^\oplus = c_i \otimes \gamma_i^\oplus$ , for  $1 \leq i \leq n$ . Then, C sends  $\delta^\oplus = \{\delta_i^\oplus\}_{1 \leq i \leq n}$  to S.

*Step 6.*

(a) S decrypts each value in  $\delta^\oplus$  and checks if the decrypted value  $\delta_i = \text{Dec}(\delta_i^\oplus)$  is zero

(b) By combining the server's vertex set  $V_S$  and the set of nonzero decrypted values  $\{\delta_i\}_{\delta_i \neq 0}$ , S obtains  $V_U$ .  $V_U$  is then sorted in ascending order and is represented as  $V_U = \{u_1, u_2, \dots, u_q\}$

*Step 7.*

(a) S uses  $V_U$  to construct an adjacency matrix  $A$  of size  $q \times q$ , where  $q$  is the number of vertex in  $V_U$ :

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,q} \\ \vdots & \ddots & \vdots \\ a_{q,1} & \cdots & a_{q,q} \end{pmatrix}. \quad (7)$$

$A$  has the property that, for each vertex pair  $u_x \in V_U$  and  $u_y \in V_U$ , if an edge exists in  $G_S$  between vertices  $u_x$  and  $u_y$ ,  $a_{x,y} = 1$ ; otherwise,  $a_{x,y} = 0$

(b) S encrypts each element in  $A$  under the Paillier cryptosystem and sends the encrypted matrix  $A^\oplus$  and  $V_U$  to C

*Step 8.*

(a) C uses  $V_U$  to construct an adjacency matrix  $B$  in the same manner as S in the last step:

$$B = \begin{pmatrix} b_{1,1} & \cdots & b_{1,q} \\ \vdots & \ddots & \vdots \\ b_{q,1} & \cdots & b_{q,q} \end{pmatrix} \quad (8)$$

(b) C encrypts each element in  $B$  using the Paillier cryptosystem and obtains  $B^\oplus$

(c) C generates a matrix  $R$  with  $q \times q$  random values:

$$R = \begin{pmatrix} r_{1,1} & \cdots & r_{1,q} \\ \vdots & \ddots & \vdots \\ r_{q,1} & \cdots & r_{q,q} \end{pmatrix} \quad (9)$$

(d) C computes:

$$\begin{aligned} E_U^\oplus &= (A^\oplus \oplus B^\oplus) \otimes R = \begin{pmatrix} (a_{1,1}^\oplus \oplus b_{1,1}^\oplus) \otimes r_{1,1} & \cdots & (a_{1,q}^\oplus \oplus b_{1,q}^\oplus) \otimes r_{1,q} \\ \vdots & \ddots & \vdots \\ (a_{q,1}^\oplus \oplus b_{q,1}^\oplus) \otimes r_{q,1} & \cdots & (a_{q,q}^\oplus \oplus b_{q,q}^\oplus) \otimes r_{q,q} \end{pmatrix} \\ &= \begin{pmatrix} e_{1,1}^\oplus & \cdots & e_{1,q}^\oplus \\ \vdots & \ddots & \vdots \\ e_{q,1}^\oplus & \cdots & e_{q,q}^\oplus \end{pmatrix} \end{aligned} \quad (10)$$

(e)  $C$  sends  $E_U^\oplus$  to  $S$

Step 9.  $S$  decrypts the matrix  $E_U^\oplus$ . For each decrypted element  $e_{i,j}$ , if  $e_{i,j} \neq 0$ , set  $e_{i,j} = 1$ . At last,  $S$  obtains  $E_U$ .

## 6. Analysis

**6.1. Security Analysis.** In this section, we prove the correctness and security of both PGI and PGU. When analyzing the security of the proposed protocols, we assume both the server and the client evaluate the protocols faithfully, but they may try to obtain as much information about the graph of the other party as possible. The security analysis for the protocols is divided into two cases, where one of the server and the client acts as the adversary in each case. Then, we prove the zero-knowledge properties of the server and the client in each case, using the methods and techniques introduced in [15, 34].

**Lemma 5** (PGI correctness). *If both participants are honest, for any  $G_S = (V_S, E_S)$  and any  $G_C = (V_C, E_C)$ , the private graph intersection protocol computes  $G_I = (V_I, E_I) = G_S \cap G_C$ .*

*Proof.* The correctness of PGI is ensured by the correctness of the FNP protocol and the homomorphic property of the Paillier cryptosystem.

During Steps 2 to 4 of the protocol, the client and the server jointly perform a FNP protocol using their vertex collections as inputs. At the end of Step 4, the server learns the vertex intersection  $V_I$ , and the client receives  $V_I$  from the server in Step 5.

In Steps 5 and 6, the server and the client construct two adjacency matrices by using  $V_I$ , denoted as  $A$  and  $B$ , respectively. Note that  $A$  and  $B$  contain the adjacency relations between the vertices in  $V_I$  for graphs  $G_S$  and  $G_C$ , respectively. In other words, if an edge exists between two vertices in  $V_I$ , it leads to a value of 1 in the corresponding position of the constructed adjacency matrix; otherwise, it leads to a value of 0 instead. Therefore, the dot product of  $A$  and  $B$  will produce an adjacency matrix that represents the edge intersection. If an edge exists in both  $A$  and  $B$ , i.e., it is a common edge between  $G_S$  and  $G_C$ , the dot product of its adjacency relations will result a value of 1. If an edge only exists in one of  $G_S$  and  $G_C$ , or the edge does not exist at all, the dot product will result in a value of 0.

In Step 6, the client receives the encryption of  $A$  under the Paillier cryptosystem from the server. If the Paillier cryptosystem has the homomorphic property, i.e., it supports multiplication between a ciphertext and a constant, the client can homomorphically compute the dot product of the  $A^\oplus$  and  $B$ , and the result is the encryption of the edge intersection. Finally, in Step 7, the server obtains the edge intersection after decryption.

As a result, if the FNP protocol is correct and the Paillier cryptosystem has the homomorphic property, the private graph intersection protocol computes  $G_I = (V_I, E_I) = G_S \cap G_C$ .

**Lemma 6** (PGI server zero-knowledge). *A semihonest server learns nothing about the client's graph, beyond what can be deduced from  $G_I$  and the leakage function  $\mathcal{L}_1$ .*

*Proof.* The proof of PGI server zero-knowledge is trivial. During PGI, there are two parts where the server receives information about the client's graph. The first part is during the FNP protocol in Step 3, and the second part is at the end of Step 6.

For the first part, in Step 3, the server receives a set of ciphertexts from the client. The server can learn the number of vertices in the client's graph by counting the number of ciphertexts, which is the predefined leakage function  $\mathcal{L}_1$ . By decrypting the ciphertexts, the server obtains a set of values. If a value exists in  $V_S$ , it is a common vertex between  $G_S$  and  $G_C$ , which is a part of the final result of the protocol. Otherwise, if the value does not exist in  $V_S$ , it will be a random value, which has no relation to the client's graph.

For the second part, the server receives  $E_I^\oplus$  from the client, which is the ciphertext of the edge intersection. Upon decryption, the server only learns the edge intersection. As a result, the PGI server zero-knowledge holds.

**Lemma 7** (PGI client zero-knowledge). *A semihonest client learns nothing about the server's graph, beyond the leakage function  $\mathcal{L}_2$ .*

*Proof.* There are two parts where the client receives information about the server's graph. The first part is during the FNP protocol in Step 2, and the second part is at the end of Step 5.

For the first part, the client receives a set of encrypted coefficients  $\alpha^\oplus$  of the polynomial  $P$  from the server. The client can learn the number of vertices of the server's graph by counting the number of encrypted coefficients received, which is a part of the predefined leakage function  $\mathcal{L}_2$ .

For the second part, the client receives an encrypted matrix  $A^\oplus$  and the vertex intersection  $V_I$ . Since  $V_I$  is also a part of the predefined leakage function  $\mathcal{L}_2$ , we need to show that  $A^\oplus$  does not reveal any information about the server's graph. According to the protocol construction,  $A^\oplus$  contains the encryptions of adjacency relations between the vertices in  $V_I$  for the server's graph. Therefore, if the client cannot distinguish between the cases where the server has different input graphs, given the knowledge of  $A^\oplus$  and  $V_I$ , the PGI client zero-knowledge holds. Consider the following experiment:

$$\begin{aligned} & \text{EXP}_{\mathcal{A}}(1^k), \\ & (G_0, G_1) \leftarrow \mathcal{A}, \\ & b \leftarrow \mathcal{S}\{0, 1\}, \\ & (pk, sk) \leftarrow \text{Step 1}(1^k), \\ & \alpha^\oplus \leftarrow \text{Step 2}(G_b, pk), \end{aligned}$$

$$\begin{aligned}
\beta^\oplus &\leftarrow \text{Step 3}(\alpha^\oplus, G_C), \\
V_I &\leftarrow \text{Step 4}(\beta^\oplus, sk), \\
A^\oplus &\leftarrow \text{Step 5}(G_b, V_I, pk), \\
\hat{b} &\leftarrow \mathcal{A}(\alpha^\oplus, V_I, A^\oplus) \text{ if } \hat{b} = b, \text{ output 1,} \\
&\text{otherwise, output 0.} \tag{11}
\end{aligned}$$

In the above experiment,  $\mathcal{A}$  is a probabilistic polynomial-time adversarial client with a private graph  $G_C = (E_C, V_C)$ . The adversary first chooses two graphs, denoted as  $G_0 = (V_0, E_0)$  and  $G_1 = (V_1, E_1)$ , respectively. The two graphs have the property that  $V_0 \cap V_C = V_1 \cap V_C$  and  $|V_0| = |V_1|$ .  $\mathcal{A}$  then sends the graphs to the server. The server randomly picks a bit  $b = \{0, 1\}$ , and chooses  $G_b$  as the private graph. After that, the server and  $\mathcal{A}$  jointly perform the private graph intersection protocol from Steps 1 to 5.

At the end of Step 5,  $\mathcal{A}$  needs to output a bit  $\hat{b}$ , using the information he received during the protocol. If  $\hat{b} = b$ , the experiment outputs 1; otherwise, it outputs 0. The advantage of the above experiment for  $\mathcal{A}$  is defined as  $\text{Adv}_{\mathcal{A}} = |\Pr [\text{EX P}_{\mathcal{A}}(1^k) = 1] - 1/2|$ .

During PGI, the information that  $\mathcal{A}$  receives contains  $\alpha^\oplus$ ,  $V_I$ , and  $A^\oplus$ .  $\alpha^\oplus$  contains a set of ciphertexts under the Paillier cryptosystem,  $V_I$  is the vertex intersection, and  $A^\oplus$  is an encrypted adjacency matrix under the Paillier cryptosystem.

Due to the condition  $V_0 \cap V_C = V_1 \cap V_C$ , the vertex intersection  $V_I$  gives no useful information since  $V_I$  will be the same for both  $G_0$  and  $G_1$ . Since the Paillier cryptosystem is IND-CPA secure and  $\mathcal{A}$  cannot decrypt the ciphertexts without the private key,  $\alpha^\oplus$  and  $A^\oplus$  cannot help  $\mathcal{A}$  to distinguish which graph the server has chosen. As a result, if the Paillier cryptosystem is IND-CPA secure, the advantage of the above experiment for  $\mathcal{A}$  is negligible, i.e.,  $\text{Adv}_{\mathcal{A}} = |\Pr [\text{EXP}_{\mathcal{A}}(1^k) = 1] - 1/2| = \epsilon$ , where  $\epsilon$  is negligible.

At last, we construct a simulator  $\text{Sim}_S$  to simulate the view of the client in the ideal model.  $\text{Sim}_S$  is given the knowledge of the vertex intersection  $V_I$  and the vertex number  $m$  of the server's graph. In the above experiment,  $\text{Sim}_S$  sends a set of  $m + 1$  random values to the client in Step 2 and sends  $V_I$  and a matrix with  $p \times p$  random values to the client in Step 5. Since the client cannot distinguish between the ciphertexts under the Paillier cryptosystem and random values, the view of the client in the ideal model is computationally indistinguishable from the view in the real model, i.e.,  $\text{View}_C^{\text{real}}[S(G_S), C] \approx \text{View}_C^{\text{ideal}}[\text{Sim}_S(V_I, m), C]$ . As a result, the PGI client zero-knowledge holds.

**Lemma 8** (PGU correctness). *If both participants are honest, for any  $G_S = (V_S, E_S)$  and any  $G_C = (V_C, E_C)$ , the private graph union protocol computes  $G_U = (V_U, E_U) = G_S \cup G_C$ .*

*Proof.* The correctness of PGU is ensured by the homomorphic property of the Paillier cryptosystem. Steps 2– of PGU

compute the vertex union, and Steps 7–9 compute the edge union.

In order to compute the vertex union between  $G_S$  and  $G_C$ , the server needs to obtain the vertices in  $G_C$  that are not in  $G_S$ .

In Step 2, the server constructs a polynomial, such that all the roots are exactly the vertices in  $G_S$ . After that, the client homomorphically evaluates the polynomial using all the vertices in  $G_C$ , and each polynomial evaluation is homomorphically multiplied by a random value. Therefore, the common vertices between  $G_S$  and  $G_C$  will result in encryptions of zero, and other vertices will result in encryptions of random values. In Step 4, the server decrypts all the polynomial evaluations. If the decryption is zero, the server generates an encryption of 0; otherwise, the server generates an encryption of 1. In the next step, the client homomorphically multiplies the received encryptions with the vertices in  $V_C$ . For an encryption of 0, i.e., the vertex is a common vertex, the client will result in an encryption of 0; for an encryption of 1, i.e., the vertex is not a common vertex, the client will result in an encryption of the vertex. As a result, in Step 6, the server learns the set of vertices that only exists in  $G_C$ . By combing the above set and  $V_S$ , the server obtains the vertex union  $V_U$ .

In order to compute the edge union, the server needs to obtain an adjacency matrix, such that if an edge does not exist in either  $G_S$  and  $G_C$ , it will have a corresponding value of 0 in the matrix; otherwise, it will have a corresponding value of 1.

In Steps 7 and 8, each of the server and the client constructs an adjacency matrix using the vertex union and his own graph and encrypts each element under the Paillier cryptosystem. The client then homomorphically adds the encrypted values at the same locations in the two matrices. There are three circumstances for the addition results. If an edge does not exist in either of the graphs, the addition will result in an encryption of 0; if an edge only exists in one of the graphs, the addition will result in an encryption of 1; if an edge exists in both of the graphs, the addition will result in an encryption of 2. Then, the client homomorphically multiplies each result by a random value. Therefore, for the edges that do not exist in either of the graphs, the final result will still be an encryption of 0; for the edges that only exist in one of the graphs and the edges that exist in both of the graphs, the final result will be encryptions of random values. Finally, in Step 9, the server decrypts the encrypted matrix and replaces all the nonzero values to 1, which is the edge union of  $G_S$  and  $G_C$ .

As a result, if the Paillier cryptosystem has the homomorphic property, the private graph union protocol computes  $G_U = (V_U, E_U) = G_S \cup G_C$ .

**Lemma 9** (PGU server zero-knowledge). *A semihonest server learns nothing about the client's graph, beyond what can be deduced from  $G_U$  and the leakage function  $\mathcal{L}_3$ .*

*Proof.* There are three parts where the server receives information from the client, which are Steps 3, 5, and 8.

In Step 3, the server receives a set of ciphertexts,  $\beta^\oplus$ , from the client. Each vertex in  $V_C$  has a corresponding ciphertext in  $\beta^\oplus$ . If a vertex in  $V_C$  also exists in  $V_S$ , i.e., it is a common vertex in both graphs, it will result in an encryption of 0; otherwise, it will result in an encryption of a random value. By counting the number of ciphertexts in  $\beta^\oplus$ , the server can learn the number of vertices in the client's graph, and by decrypting and counting the number of 0s, the server can learn the number of common vertices. The above information is defined as leakage function  $\mathcal{L}_3$ .

In Step 5, the server receives another set of ciphertexts,  $\gamma^\oplus$ , from the client. Similar as above, each vertex in  $V_C$  has a corresponding ciphertext in  $\gamma^\oplus$ . If a vertex exists in both  $V_S$  and  $V_C$ , it will result in an encryption of 0; otherwise, it will result in an encryption of the vertex itself. Therefore, upon decryption, the server learns of the vertices in  $V_C$  that do not exist in  $V_S$ , which are a part of the vertex union.

In Step 8, the server receives an encrypted matrix,  $E_U^\oplus$ , from the client. Each element of the matrix represents the adjacency relation between two vertices in the graph union. If an edge exists in at least one of the input graphs, the corresponding adjacency value will be a random number; if an edge does not exist in either of the input graphs, it will result in an adjacency value of 0. By decrypting the matrix and replacing the random values to 1, the server obtains the edge union. As a result, the PGU server zero-knowledge holds.

**Lemma 10** (PGU client zero-knowledge). *A semihonest client learns nothing about the server's graph, beyond what can be deduced from  $V_U$  and the leakage function  $\mathcal{L}_4$ .*

*Proof.* There are three parts where the client receives information from the server, which are Steps 2, 4, and 7. In Step 2, the client receives a set  $\alpha^\oplus$  that contains  $m + 1$  ciphertexts under the Paillier cryptosystem, which are encryptions of the coefficients of the server's polynomial. The client can learn the vertex number of the server's graph by counting the ciphertexts in  $\alpha^\oplus$ , which is the leakage function  $\mathcal{L}_4$ . In Step 4, the client receives another set of ciphertexts  $\gamma^\oplus$ , which contains  $n$  encryptions of 1s and 0s. In Step 7, the client receives an encrypted matrix of size  $q \times q$ , which contains encryptions of 1s and 0s. In order to prove that the above information does not reveal anything about the server's graph beyond what can be deduced from  $V_U$  and the leakage function  $\mathcal{L}_4$ , consider the following experiment:

$$\begin{aligned} & \text{EXP}_{\mathcal{A}}(1^k), \\ & (G_0, G_1) \leftarrow \mathcal{A}, \\ & b \leftarrow \mathcal{S}\{0, 1\}, \\ & (pk, sk) \leftarrow \text{Step 1}(1^k), \\ & \alpha^\oplus \leftarrow \text{Step 2}(G_b, pk), \\ & \beta^\oplus \leftarrow \text{Step 3}(\alpha^\oplus, G_C), \\ & \gamma^\oplus \leftarrow \text{Step 4}(\beta^\oplus, pk, sk), \end{aligned}$$

$$\delta^\oplus \leftarrow \text{Step 5}(\gamma^\oplus, G_C),$$

$$V_U \leftarrow \text{Step 6}(\delta^\oplus, sk, G_b),$$

$$A^\oplus \leftarrow \text{Step 7}(G_b, V_U, pk),$$

$$\hat{b} \leftarrow \mathcal{A}(\alpha^\oplus, \gamma^\oplus, A^\oplus, V_U), \text{if } \hat{b} = b, \text{ output 1,}$$

$$\text{otherwise, output 0.} \quad (12)$$

In the above experiment,  $\mathcal{A}$  is a probabilistic polynomial-time adversarial client with a private graph  $G_C = (E_C, V_C)$ . The adversary first chooses two graphs, denoted as  $G_0 = (V_0, E_0)$  and  $G_1 = (V_1, E_1)$ , respectively. The two graphs have the property that  $V_0 \cup V_C = V_1 \cup V_C$  and  $|V_0| = |V_1|$ .  $\mathcal{A}$  then sends the graphs to the server. The server randomly picks a bit  $b = \{0, 1\}$  and chooses  $G_b$  as the private graph. After that, the server and  $\mathcal{A}$  jointly perform the private graph union protocol from Steps 1 to 7.

At the end of Step 7,  $\mathcal{A}$  needs to output a bit  $\hat{b}$ , using the information he received during the protocol. If  $\hat{b} = b$ , the experiment outputs 1; otherwise, it outputs 0. The advantage of the above experiment for  $\mathcal{A}$  is defined as  $\text{Adv}_{\mathcal{A}} = |\Pr[\text{EXP}_{\mathcal{A}}(1^k) = 1] - 1/2|$ .

During PGU, the information that  $\mathcal{A}$  receives contains  $\alpha^\oplus, \gamma^\oplus, A^\oplus$ , and  $V_U$ .  $\alpha^\oplus$  and  $\gamma^\oplus$  are both sets of ciphertexts under the Paillier cryptosystem. Since  $G_0$  and  $G_1$  satisfied the condition  $|V_0| = |V_1|$ , the numbers of ciphertexts in  $\alpha^\oplus$  will be the same for both  $G_0$  and  $G_1$ .  $A^\oplus$  is a matrix filled with  $q \times q$  ciphertexts. Since the Paillier cryptosystem is IND-CPA secure and  $\mathcal{A}$  cannot decrypt the ciphertexts without the private key,  $\alpha^\oplus, \gamma^\oplus$ , and  $A^\oplus$  cannot help  $\mathcal{A}$  to distinguish which graph the server has chosen. Furthermore, since  $G_0$  and  $G_1$  satisfied the condition  $V_0 \cup V_C = V_1 \cup V_C$ ,  $V_U$  will be the same for both  $G_0$  and  $G_1$ . As a result, if the Paillier cryptosystem is IND-CPA secure, the advantage of the above experiment for  $\mathcal{A}$  is negligible, i.e.,  $\text{Adv}_{\mathcal{A}} = |\Pr[\text{EXP}_{\mathcal{A}}(1^k) = 1] - 1/2| = \varepsilon$ , where  $\varepsilon$  is negligible.

At last, we construct a simulator  $\text{Sim}_S$  to simulate the view of the client in the ideal model.  $\text{Sim}_S$  is given the knowledge of the vertex union  $V_U$  and the vertex number  $m$  of the server's graph. In the ideal model,  $\text{Sim}_S$  generates a set of  $m + 1$  random values in Step 2, a set of  $n$  random values in Step 4, and a matrix of size  $q \times q$  filled with random values in Step 7. Since the Paillier cryptosystem is IND-CPA secure, the client cannot distinguish the ciphertexts and random values. Therefore, the view of the client in the ideal model is computationally indistinguishable from the view in the real model, i.e.,  $\text{View}_C^{\text{real}}[S(G_S), C] \approx \text{View}_C^{\text{ideal}}[\text{Sim}_S(V_U, m), C]$ . As a result, the PGU client zero-knowledge holds.

**6.2. Efficiency Analysis.** In this section, we analyze the efficiencies of PGI and PGU in terms of communication cost and computation cost. The communication cost is measured in terms of the amount of ciphertexts that has been transferred between the server and the client, and the computation



cost is measured in terms of modular exponentiations and multiplications.

We denote  $m$  as the number of vertices in  $G_S$ ,  $n$  as the number of vertices in  $G_C$ ,  $p$  as the number of vertices in the intersection of  $G_S$  and  $G_C$ , and  $q$  as the number of vertices in the union of  $G_S$  and  $G_C$ .

**6.2.1. PGI Communication Cost.** The construction of PGI is simple and only requires  $O(1)$  rounds of communication. In Step 2, the server sends  $m + 1$  ciphertexts to the client. In Step 3, the client sends  $n$  ciphertexts to the server. In Step 5, the server sends  $p^2$  ciphertexts to the client. At last, in Step 6, the client sends  $p^2$  ciphertexts to the server. As a result, the total communication cost of our protocol is  $O(m + n + p^2)$  ciphertexts.

**6.2.2. PGI Server Computation Cost.** In Step 2, constructing the polynomial requires  $O(m^2)$  modular multiplication, and encrypting the coefficients requires  $O(m)$  modular exponentiations. In Step 4, decrypting the received ciphertexts requires  $O(n)$  modular exponentiations. In Step 5, encrypting each element in  $A$  requires  $O(p^2)$  modular exponentiations. In Step 7, decrypting each element in  $E_7^\oplus$  requires  $O(p^2)$  exponentiations. As a result, the total computation cost for the server is  $O(m + n + p^2)$  modular exponentiations and  $O(m^2)$  modular multiplications.

**6.2.3. PGI Client Computation Cost.** In Step 3, obviously evaluating the polynomial requires  $O(mn)$  modular exponentiations. In Step 6, computing  $E_6^\oplus$  requires  $O(p^2)$  modular exponentiations. As a result, the total computation cost for the client is  $O(mn + p^2)$  modular exponentiations.

**6.2.4. PGU Communication Cost.** The construction of PGU also only requires  $O(1)$  rounds of communication. During Steps 2–5, the server sends  $m + 1 + n$  ciphertexts to the client, and the client sends  $2n$  ciphertexts to the server. During Steps 7 and 8, the server sends  $q^2$  ciphertexts to the client, and the client sends  $q^2$  ciphertexts to the server. As a result, the total communication cost is  $O(m + n + q^2)$  ciphertexts.

**6.2.5. PGU Server Computation Cost.** In Step 2, constructing the polynomial requires  $O(m^2)$  modular multiplication, and encrypting the coefficients requires  $O(m)$  modular exponentiations. In Step 4, decrypting  $n$  ciphertexts requires  $O(n)$  modular exponentiations, and encrypting  $n$  ciphertexts requires  $O(n)$  modular exponentiations. In Step 6, decrypting  $n$  ciphertexts requires  $O(n)$  modular exponentiations. In Step 7, encrypting each element in  $A$  requires  $O(q^2)$  modular exponentiations. In Step 9, decrypting each element in  $E_9^\oplus$  requires  $O(q^2)$  modular exponentiations. As a result, the total computation cost for the server is  $O(m + n + q^2)$  modular exponentiations and  $O(m^2)$  modular multiplications.

**6.2.6. PGU Client Computation Cost.** In Step 3, obviously evaluating the polynomial requires  $O(mn)$  modular exponentiations. Computing  $n$  homomorphic multiplication requires  $O(n)$  modular exponentiations. In Step 5, computing  $n$  homomorphic multiplication requires  $O(n)$  modular

exponentiations. In Step 8, encrypting the each element in  $B$  requires  $O(q^2)$  modular exponentiations. Computing  $q^2$  homomorphic addition and multiplication requires  $O(q^2)$  modular exponentiations and  $O(q^2)$  modular multiplication. As a result, the total computation cost for the client is  $O(m + n + q^2)$  modular exponentiations and  $O(q^2)$  modular multiplications.

### 6.3. Leakage Analysis

**6.3.1. PGI Leakage.** As stated before, the proposed PGI leaks certain information about the private graphs, which is modeled as the leakage functions  $\mathcal{L}_1$  and  $\mathcal{L}_2$ . There are several techniques that can be used to reduce the amount of information leakage; however, it cannot be completely avoided.

In Step 2, the server constructs a polynomial  $P$ , such that all the roots of  $P(x)$  are exactly the elements in  $V_S$ . After that, the server sends the encryptions of the coefficients of  $P$  to the client. In order to prevent the client from learning the exact vertex number of the server's graph, the server first randomly constructs an irreducible polynomial  $R(x)$  with degree  $d$ . The server then computes  $P'(x) = P(x)R(x)$  and uses  $P'(x)$  instead of  $P(x)$  in Step 2. The polynomial  $P'(x)$  has the same property as  $P(x)$ ; therefore, it will not affect the result of the protocol. As a result, by counting the number of ciphertexts received, the client can only learn the upper bound of the vertex number of the server's graph, i.e.,  $m + d$ .

In order to hide the exact vertex number of the client's graph, the client can randomly generate a set of  $h$  values from the message space of the Paillier cryptosystem in Step 3. After that, the client encrypts the random values and sends the encrypted random set to the server along with  $\beta^\oplus$ . Since the message space of the Paillier cryptosystem is large enough, the probability that a random value equals to an element in  $V_S$  can be assumed as negligible. Therefore, the random values will not affect the result of the protocol, since they are not in the vertex intersection. As a result, by counting the number ciphertexts received in Step 3, the server can only learn the upper bound of the vertex number of the client's graph, i.e.,  $n + h$ .

**6.3.2. PGU Leakage.** Similar as PGI, PGU also leaks partial information about the input graphs during the process, which is modeled as  $\mathcal{L}_3$  and  $\mathcal{L}_4$ .

In Step 2, the server can utilize the same technique, as introduced above, to hide the exact vertex number of his graph, and the client can only learn the upper bound instead, i.e.,  $m + d$ .

In Step 3, in order to hide the exact vertex number of the client's graph, the client generates  $k$  encryptions of zero and sends the ciphertexts along with  $\beta^\oplus$ . An encryption of zero in Step 3 indicates that a vertex in the client's graph also exists in the server's graph. In later steps, extra encryptions of zero will not affect the final result, since the vertex union between the two input graphs will remain the same. As a result, the server can only learn the upper bound of the vertex number of the client's graph, i.e.,  $n + k$ , and the upper bound of the common vertex number, i.e.,  $p + k$ .

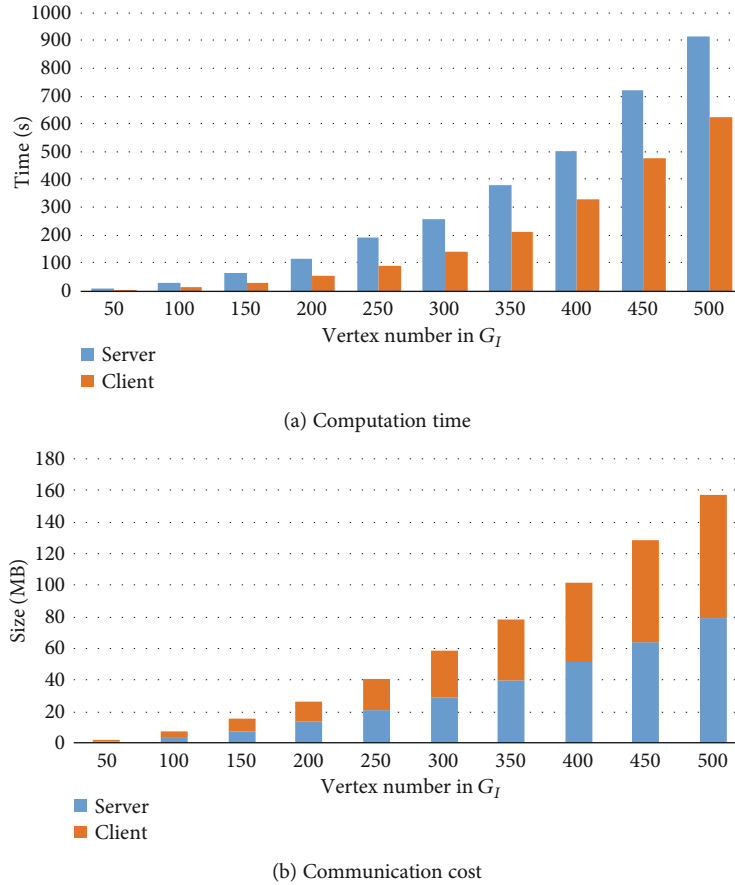


FIGURE 2: Evaluation of private graph intersection protocol.

In addition, we consider the case where the server sends a graph with small size to the client in Step 2. If the server’s graph is small enough, i.e., only 1 vertex and no edge, the union of the graphs will be almost the graph of the client. To prevent the server from learning the client’s graph in such a method, there are two points where the client can choose to end the protocol.

The first point is at Step 3. If the client receives a very small polynomial, the client can choose to end the protocol, and at this point, the server has not learned anything yet. However, if the server uses the technique stated above, the polynomial that the client receives will not give the exact size of the server’s graph. In this case, the client can check if the vertex union received in Step 8 is almost the same as his vertex set  $V_C$ . If  $V_U \approx V_C$ , it means either the server has a very small graph or the vertices in both graphs are highly overlapping. At this point, the client can choose to end the protocol; however, the server has already learned the vertex set of the client.

## 7. Experiments

In order to evaluate the performances of the proposed PGI and PGU protocols, we implement the protocols and perform experiments over the Enron email dataset. All the experiments were conducted on two PCs with Intel Core i7-2600 4.2 GHz CPU, 16 GB RAM, and Windows 10 operat-

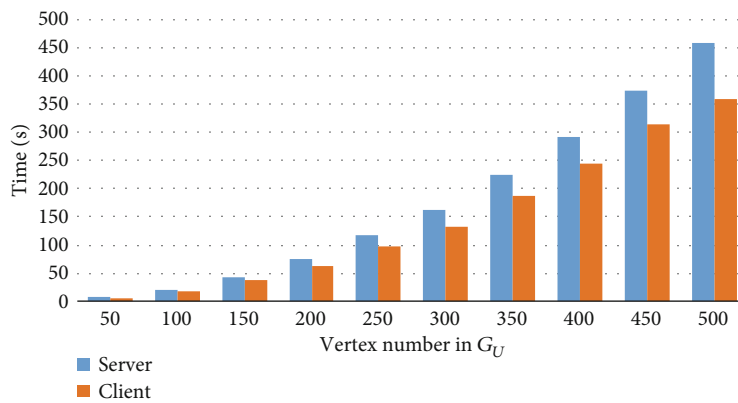
ing system. (Due to the COVID-19 crisis, we cannot access the lab in the university at the moment, which contains the environment and equipment to perform the experiments on real mobile devices. As a result, the experiments are performed on a PC in this paper, and we will improve the experiments on mobile devices in later works.). The protocols are implemented in Python 3.6, and we used the phe library for the Paillier cryptosystem with a 1024-bit key length.

**7.1. Dataset.** The Enron email dataset is publicly available from the Stanford SNAP website (<https://snap.stanford.edu/data/>). The dataset contains email communications of around half a million emails. In order to convert the dataset to a graph, the senders and the receivers of the emails are represented as vertices, and if vertex  $i$  sends at least one email to vertex  $j$ , there exists an undirected edge between  $i$  and  $j$ . The resulting graph has 36,692 vertices and 183,831 edges. In addition, each vertex of the graph is represented as a unique integer.

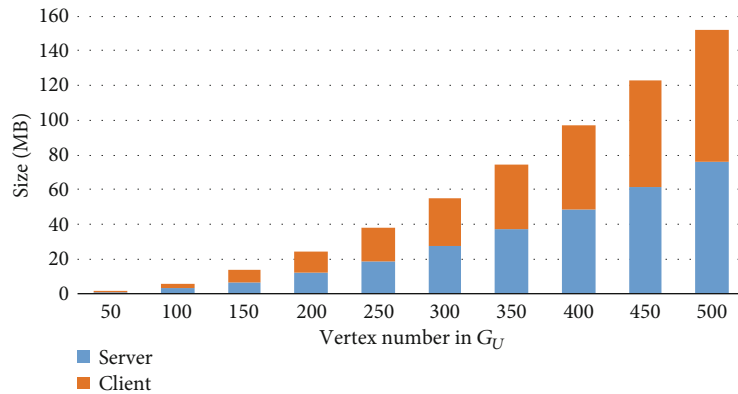
**7.2. Evaluation of PGI.** When evaluating the performance of PGI, we randomly generate two subgraphs from the Enron email graph dataset and assign them to the server and the client, respectively. For each experiment, we set  $m$  and  $n$  to have the same value, and they increase from 1,000 to 10,000. Furthermore, the graphs of the server and the client are generated following the rule that 5% of the vertices are the same

TABLE 2: The computation time for each step of private graph intersection protocol.

$m/n$	$p$	Step 2 time (s)	Step 3 time (s)	Step 4 time (s)	Step 5 time (s)	Step 6 time (s)	Step 7 time (s)
1,000	50	2.07	3.26	1.05	4.72	1.04	1.22
2,000	100	3.58	6.17	1.90	20.26	5.01	2.94
3,000	150	4.90	9.79	2.12	48.16	16.57	6.15
4,000	200	6.57	15.96	2.64	95.95	38.40	10.49
5,000	250	7.93	24.42	3.38	164.47	62.32	14.35
6,000	300	8.68	32.85	3.74	225.50	108.21	20.23
7,000	350	10.07	46.65	4.70	336.07	165.52	26.87
8,000	400	11.14	65.60	5.54	449.29	262.08	36.90
9,000	450	13.35	96.32	6.63	654.43	382.28	46.88
10,000	500	15.48	130.69	8.19	832.97	493.01	58.28



(a) Computation Time



(b) Communication Cost

FIGURE 3: Evaluation of private graph union protocol.

between the two graphs. Figure 2(a) shows the computation time for the server and the client.

As analyzed before, the computation costs for the server and the client are  $O(m + n + p^2)$  and  $O(mn + p^2)$ , respectively, where  $p$  is the number of vertices in the intersection of  $G_S$  and  $G_C$ . Therefore, the most dominant part of the computation costs for both the server and the client is most likely to be the number of common vertices between  $G_S$  and  $G_C$ . As shown in Figure 2(a), the computation time for both the server and the client grows quadratically as the number of

common vertices increases. The detailed computation time for each step is shown in Table 2.

As shown in Table 2, the most time-consuming parts of PGI are Steps 5 and 6. In Step 5, the server performs  $p^2$  Paillier encryptions, and in Step 6, the client performs  $p^2$  homomorphic multiplications. Since the computations for both Steps 5 and 6 are highly parallelizable, the computation time can be greatly reduced if cluster computing is deployed.

The communication costs of PGI for both the server and the client are shown in Figure 2(b). As analyzed before, the

TABLE 3: The computation time for each step of private graph union protocol.

$m/n$	$q$	Step 2 time (s)	Step 3 time (s)	Step 4 time (s)	Step 5 time (s)	Step 6 time (s)	Step 7 time (s)	Step 8 time (s)	Step 9 time (s)
30	50	0.54	1.13	0.52	0.45	0.46	4.10	5.12	1.74
60	100	0.59	1.23	0.50	0.50	0.52	14.32	16.62	5.43
90	150	0.66	1.32	0.47	0.47	0.51	31.17	35.66	10.94
120	200	0.68	1.26	0.45	0.45	0.53	54.90	62.11	19.25
150	250	0.77	1.30	0.48	0.48	0.52	84.50	96.92	30.05
180	300	0.80	1.32	0.52	0.52	0.65	117.55	130.72	43.26
210	350	0.82	1.38	0.48	0.48	0.63	162.46	185.12	59.09
240	400	0.80	1.48	0.49	0.49	0.65	212.23	242.75	77.69
270	450	0.95	1.59	0.58	0.58	0.65	275.12	311.50	94.98
300	500	0.87	1.52	0.53	0.53	0.63	336.23	355.94	118.34

total communication cost is  $O(m + n + p^2)$ . As a result, the communication costs have a quadratic growth in the figure. In addition, the communication costs are nearly the same for both the server and the client, and the overall communication cost for PGI is practical for the experimental dataset.

**7.3. Evaluation of PGU.** When evaluating the performance of PGU, we first randomly generate a subgraph from the Enron email graph dataset as the graph union  $G_U$ . Then, we randomly choose two subgraphs of  $G_U$  and assign them to the server and the client, respectively. The numbers of the vertices in the subgraphs are 60% of the vertex number in  $G_U$ ; therefore, both  $m$  and  $n$  will have the same value. For each experiment, the number of vertices in  $G_U$  increases from 50 to 500. Figure 3(a) shows the computation time for the server and the client, and the detailed computation time for each step is shown in Table 3.

As analyzed before, the computation costs for PGU are  $O(m + n + q^2)$  and  $O(mn + q^2)$  for the server and the client, respectively, where  $q$  is the number of vertices in the union of  $G_S$  and  $G_C$ . Therefore, similar as PGI, the most dominant part of the computation costs for both the server and the client is most likely to be the number of vertices in  $G_U$ .

As shown in Table 3, most of the computation time is spent in Steps 7 and 8. In Step 7, the server performs  $q^2$  Paillier encryptions, and in Step 8, the client performs  $q^2$  Paillier encryptions and  $q^2$  homomorphic additions and multiplications. Similar as before, the above computations are highly parallelizable, and cluster computing will greatly optimize the computation time.

As shown in Figure 3(b), the communication cost of PGU is similar to PGI, and the communication costs for both the server and the client have a quadratic growth as the number of vertices in  $G_U$  increases. For our experimental dataset, the overall communication cost for the PGU protocol is also practical.

## 8. Conclusion

In this work, we proposed two privacy-preserving graph operation protocols between two parties, which can be used for secure authentication for smart devices. The first protocol, PGI, allows a server and a client to jointly compute the

intersection between their private graphs, while the second protocol, PGU, computes the union of the graphs. The protocols first use polynomial representation and oblivious polynomial evaluation to compute the intersection and union of the vertices. The intersection and union of the edges are then computed by using an additive homomorphic cryptosystem.

We proved that the proposed protocols are secure in the semihonest security model. In other words, a semihonest client learns nothing about the server's graph and a semihonest server learns nothing about the client's graph. We analyzed the leakages during the protocols for both the server and the client and modeled the leakages as leakage functions. At last, we implemented the constructions of the protocols and evaluated the efficiencies over real-word graph data.

## Data Availability

The graph data used to support the findings of this study can be found at <https://snap.stanford.edu/data/>.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (61872069) and the Fundamental Research Funds for the Central Universities (N2017012).

## References

- [1] M. A. Khan and K. Salah, "IoT security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.
- [2] Z. K. Zhang, M. C. Y. Cho, C. W. Wang, C. W. Hsu, C. K. Chen, and S. Shieh, "IoT security: ongoing challenges and research opportunities," in *2014 IEEE 7th international conference on service-oriented computing and applications*, pp. 230–234, Matsue, Japan, 2014.
- [3] M. El-hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of internet of things (IoT) authentication schemes," *Sensors*, vol. 19, no. 5, p. 1141, 2019.

- [4] M. Wazid, A. K. Das, R. Hussain, G. Succi, and J. J. Rodrigues, "Authentication in cloud-driven IoT-based big data environment: survey and outlook," *Journal of Systems Architecture*, vol. 97, pp. 185–196, 2019.
- [5] I. H. Chuang, B. J. Guo, J. S. Tsai, and Y. H. Kuo, "Multi-graph zero-knowledge-based authentication system in internet of things," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, 2017.
- [6] C. Lin, D. He, X. Huang, M. K. Khan, and K. K. R. Choo, "A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems," *IEEE Access*, vol. 6, pp. 28203–28212, 2018.
- [7] Z. Shao, Z. Li, P. Wu, L. Chen, and X. Zhang, "Multi-factor combination authentication using fuzzy graph domination model," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 4, pp. 4979–4985, 2019.
- [8] S. S. Sonawane and P. A. Kulkarni, "Graph based representation and analysis of text document: a survey of techniques," *International Journal of Computer Applications*, vol. 96, no. 19, pp. 1–8, 2014.
- [9] T. Washio and H. Motoda, "State of the art of graph-based data mining," *Acm Sigkdd Explorations Newsletter*, vol. 5, no. 1, pp. 59–68, 2003.
- [10] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 974–983, New York, USA, 2018.
- [11] S. Aridhi and E. M. Nguifo, "Big graph mining: frameworks and techniques," *Big Data Research*, vol. 6, pp. 1–10, 2016.
- [12] H. Rong, T. Ma, M. Tang, and J. Cao, "A novel subgraph  $k^+$ -isomorphism method in social network based on graph similarity detection," *Soft Computing*, vol. 22, no. 8, pp. 2583–2601, 2018.
- [13] E. Ko, M. Kang, H. J. Chang, and D. Kim, "Graph-theory based simplification techniques for efficient biological network analysis," in *2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 277–280, San Francisco, CA, USA, 2017.
- [14] V. Mukhin, Y. Romanenkov, J. Bilokin et al., "The method of variant synthesis of information and communication network structures on the basis of the graph and set-theoretical models," *International Journal of Intelligent Systems and Applications*, vol. 9, no. 11, pp. 42–51, 2017.
- [15] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K. K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9390–9401, 2020.
- [16] F. Zhou, Z. Xu, Y. Li, J. Xu, and S. Peng, "Private graph intersection protocol," in *Australasian Conference on Information Security and Privacy*, pp. 235–248, Springer, 2017.
- [17] Y. Jia, Y. Xiao, J. Yu, X. Cheng, Z. Liang, and Z. Wan, "A novel graph-based mechanism for identifying Trac vulnerabilities in smart home IoT," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 1493–1501, Honolulu, HI, USA, 2018.
- [18] F. Zhu, W. Wu, Y. Zhang, and X. Chen, "Privacy-preserving authentication for general directed graphs in industrial IoT," *Information Sciences*, vol. 502, pp. 218–228, 2019.
- [19] S. Micali and R. L. Rivest, "Transitive signature schemes," in *Cryptographers' Track at the RSA Conference, Privacy-preserving Graph Operations for Mobile Authentication*, pp. 236–243, Springer, 2002.
- [20] A. C. C. Yao, "Protocols for secure computations," in *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, vol. 82, pp. 160–164, Chicago, IL, USA, 1982.
- [21] O. Goldreich, S. Micali, and A. Wigderson, *How to play any mental game or a completeness theorem for protocols with honest majority*, STOC, 1987.
- [22] M. Blanton and E. Aguiar, "Private and oblivious set and multiset operations," *International Journal of Information Security*, vol. 15, no. 5, pp. 493–518, 2016.
- [23] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1945–1956, 2018.
- [24] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, Springer, 1999.
- [25] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *International conference on the theory and applications of cryptographic techniques*, pp. 1–19, Springer, 2004.
- [26] H. Chen, K. Laine, and P. Rindal, "Fast private set intersection from homomorphic encryption," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1243–1255, New York, USA, 2017.
- [27] B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai, "Spot-light: lightweight private set intersection from sparse OT extension," in *Annual International Cryptology Conference*, pp. 401–431, Springer, 2019.
- [28] B. Pinkas, T. Schneider, and M. Zohner, "Scalable private set intersection based on OT extension," *ACM Transactions on Privacy and Security (TOPS)*, vol. 21, no. 2, pp. 1–35, 2018.
- [29] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions," *Computer Networks*, vol. 73, pp. 41–57, 2014.
- [30] X. Zhang, X. Chen, J. K. Liu, and Y. Xiang, "DeepPAR and DeepDPA: privacy preserving and asynchronous deep learning for industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2081–2090, 2019.
- [31] Q. Jiang, Y. Qian, J. Ma, X. Ma, Q. Cheng, and F. Wei, "User centric three-factor authentication protocol for cloud-assisted wearable devices," *International Journal of Communication Systems*, vol. 32, no. 6, article e3900, 2019.
- [32] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, 2020.
- [33] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.
- [34] D. Wang, D. He, P. Wang, and C. H. Chu, "Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2014.

## Research Article

# Security Analysis on “Anonymous Authentication Scheme for Smart Home Environment with Provable Security”

Meijia Xu <sup>1</sup>, Qiyong Dong <sup>1</sup>, Mai Zhou <sup>2</sup>, Chenyu Wang <sup>3</sup>, and Yangyang Liu <sup>4</sup>

<sup>1</sup>College of Cyber Science, Nankai University, Tianjin 300350, China

<sup>2</sup>School of EECS, Peking University, Beijing 100089, China

<sup>3</sup>School of CyberSpace security, Beijing University of Posts and Telecommunications, China

<sup>4</sup>China Academy of Information and Communications Technology, China

Correspondence should be addressed to Chenyu Wang; wangchenyu@bupt.edu.cn

Received 11 August 2020; Revised 17 September 2020; Accepted 16 October 2020; Published 16 November 2020

Academic Editor: Qi Jiang

Copyright © 2020 Meijia Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As an important application of the Internet of Things, smart home has greatly facilitated our life. Since the communication channels of smart home are insecure and the transmitted data are usually sensitive, a secure and anonymous user authentication scheme is required. Numerous attempts have been taken to design such authentication schemes. Recently, Shuai et al. (Computer & Security 86(2019):132146) designed an anonymous authentication scheme for smart home using elliptic curve cryptography. They claimed that the proposed scheme is secure against various attacks and provides ideal attributes. However, we show that their scheme cannot resist inside attack and offline dictionary attack and also fails to achieve forward secrecy. Furthermore, we give some suggestions to enhance the security of the scheme. These suggestions also apply to other user authentication schemes with similar flaws.

## 1. Introduction

Smart home is a new paradigm of the Internet of Things, which can greatly facilitate our life; thus, it attracts much attention. In smart home environments, the smart devices can communicate and cooperate with each other to provide comprehensive services for users. However, the conversations between the users and the smart devices are carried out in an insecure open channel. The adversary can eavesdrop the sensitive data transmitted over the insecure channel. Therefore, it is of importance to provide a security mechanism to secure the conversations. Multifactor user authentication [1, 2] is one of the important ways to identify the authenticity of a user. In a multifactor user authentication scheme for smart home environment, there are usually four participants: a set of users, the register center, the gateways, and the sensor nodes. The user owns her personal secrecy information, such as a password and a smart device. All participants are required to register in the register center. When a user wants to access real-time data stored on a sensor node,

she can initiate an access request. Then, the gateway and the sensor node will verify the user. If the user is valid, a session key will be built to encrypt the subsequent conversations. In such schemes, the adversary is usually assumed to be able to [3] (1) control the open channel, that is, she can intercept, modify, and eavesdrop the messages in the open channel; (2) list all the items in the space of passwords and identities; (3) compromise  $n - 1$  factor(s) of a  $n$ -factor authentication scheme; (4) acquire the long-term secret key when accessing forward secrecy; (5) break some of sensor nodes; (6) obtain the previous session keys; and (7) register as a legitimate participant.

Recently, numerous user authentication schemes are proposed [4–7]. Most recently, Shuai et al. [8] designed a new anonymous authentication scheme for a smart home environment. They employ the elliptic curve cryptography to authenticate the users with resistance to offline dictionary attack and generate pseudoidentity  $DID_i$  to provide user anonymity. However, some subtleties are overlooked, which results in vulnerability to various attacks. In this paper, we

demonstrate that their scheme cannot resist offline dictionary attack and inside attack and fails to achieve forward secrecy. Besides, we also discuss the causes and countermeasures of these security flaws. The countermeasures we proposed can also be applied to other authentication schemes with similar problems.

## 2. Review of Shuai et al.'s Scheme

In this section, we briefly review Shuai et al.'s scheme. The notations and abbreviations are shown in Table 1. Firstly, the registration authority RA chooses an elliptic curve  $E$  and an additive group  $G$  of  $E$  with order  $q$  and generator  $P$ . Next, RA generates a pair of private/public key  $(x, X)$ , where  $x \in Z_q^*$  and  $X = x \cdot P$ , a long-term secret key  $K$  and a hash function  $h(\cdot): \{0, 1\}^* \rightarrow Z_q^*$ . Note that  $x$  and  $K$  will be stored in GWN, and  $\{E(F_p), G, P, X, h(\cdot)\}$  will be published to all participants.

### 2.1. User Registration Phase

*Step 1.*  $U_i \Rightarrow RA : \{ID_i, HPW_i\}$ , where  $HPW_i = h(PW_i \| a)$  and  $a$  is a random nonce.

*Step 2.*  $RA \Rightarrow U_i : \{A_i, TEMP\}$ .

RA first checks the availability of  $ID_i$  and computes  $K_{GU} = h\{ID_i \| K\}$ ,  $A_1 = K_{GU} \oplus HPW_i$ . Finally, RA generates TEMP where TEMP is initialized to 0.

*Step 3.*  $U_i$  computes  $A_2 = a \oplus h(ID_i \| PW_i)$ ,  $A_3 = h(ID_i \| HPW_i)$  and stores  $\{A_1, A_2, A_3, TEMP\}$  into the mobile device.

### 2.2. The Smart Device Registration Phase

*Step 1.*  $SD_k \Rightarrow RA : \{SID_k\}$ .

*Step 2.*  $RA \Rightarrow SD_k : K_{GS}$ . RA checks the validity of  $SID_k$  and computes  $K_{GS} = h(SID_k \| K)$ .

*Step 3.*  $SD_k$  stores  $K_{GS}$ .

### 2.3. Login and Authentication Phase

*Step 1.*  $U_i \rightarrow GWN : \{DID_i, A_4, M_1, V_1\}$ .

$U_i$  provides  $ID_i$  and  $PW_i$ , and then, the mobile device computes  $a^* = A_2 \oplus h(ID_i \| PW_i)$ ,  $HPW^* = h(PW_i \| a^*)$ .  $A_3^* = h(ID_i \| HPW_i^*)$ . If  $A_3^* \neq A_3$ , the mobile device rejects the request and sets TEMP to TEMP + 1. Once  $TEMP \geq 3$ , the mobile device will be suspended till  $U_i$  reregisters. Otherwise, the mobile device computes  $K_{GU} = A_1 \oplus HPW_i$ ,  $A_4 = \omega \cdot P$ ,  $A_5 = \omega \cdot X$ ,  $DID_i = ID_i \oplus A_5$ ,  $M_1 = (R_1 \| SID_k) \oplus K_{GU}$ , and  $V_1 = h(ID_i \| R_1 \| K_{GU} \| M_1)$ , where  $R_1$  and  $\omega \in Z_n^*$  are two random numbers, and  $SID_k$  is the identity of the target  $SD_k$ .

*Step 2.*  $GWN \rightarrow SD_k : \{M_2, V_2\}$ .

TABLE 1: Notations and abbreviations.

Symbol	Description
$U_i$	$i^{\text{th}}$ user
GWN	The gateway node
$SD_k$	$j^{\text{th}}$ smart device
$ID_i$	Identity of $U_i$
$PW_i$	Password of $U_i$
$GID_j$	Identity of GWN
$SID_k$	Identity of $SD_k$
RA	Registration authority
$K$	The secret key of GWN
$\oplus$	Bitwise XOR operation
$\ $	Concatenation operation
$h(\cdot)$	One-way hash function
$\rightarrow$	A common channel
$\Rightarrow$	A secure channel

GWN computes  $A_5^* = x \cdot A_4$ ,  $ID_i^* = DID_i \oplus A_5^*$ ,  $K_{GU} = h\{ID_i^* \| K\}$ ,  $R_1^* \| SID_k = M_1 \oplus K_{GU}$ ,  $V_1^* = h(ID_i \| R_1 \| K_{GU} \| M_1)$ . If  $V_1^* \neq V_1$ , GWN ends the session. Otherwise, GWN computes  $K_{GS} = h(SID_k \| K)$ ,  $M_2 = (ID_i \| GID_j \| R_1 \| R_2) \oplus K_{GS}$ , and  $V_2 = h(ID_i \| GID_j \| K_{GS} \| R_1 \| R_2)$ , where  $R_2$  is a random number.

*Step 3.*  $SD_k \rightarrow GWN : \{M_3, V_3\}$ .

$SD_k$  computes  $(ID_i \| GID_j \| R_1 \| R_2) = M_2 \oplus K_{GS}$ ,  $V_2^* = h(ID_i \| GID_j \| K_{GS} \| R_1 \| R_2)$ . If  $V_2^* \neq V_2$ ,  $SD_k$  ends the session. Otherwise,  $SD_k$  computes  $SK = h(ID_i \| GID_j \| SID_k \| R_1 \| R_2 \| R_3)$ ,  $M_3 = R_3 \oplus K_{GS}$ , and  $V_3 = h(R_3 \| K_{GS} \| SK)$ , where  $R_3$  is a random number.

*Step 4.*  $GWN \rightarrow U_i : \{M_4, V_4\}$ .

GWN computes  $R_3 = M_3 \oplus K_{GS}$ ,  $SK = h(ID_i \| GID_j \| SID_k \| R_1 \| R_2 \| R_3)$ , and  $V_3^* = h(R_3 \| K_{GS} \| SK)$ . If  $V_3^* \neq V_3$ , GWN ends the session. Otherwise, GWN computes  $M_4 = (GID_j \| R_2 \| R_3) \oplus K_{GS}$  and  $V_4 = h(K_{GU} \| SK \| R_2 \| R_3)$ .

*Step 5.*  $U_i$  computes  $(GID_j \| R_2 \| R_3) = M_4 \oplus K_{GU}$ ,  $SK = h(ID_i \| GID_j \| SID_k \| R_1 \| R_2 \| R_3)$ , and  $V_4^* = h(K_{GU} \| SK \| R_2 \| R_3)$ . If  $V_4^* = V_4$ , the authentication is finished successfully.

## 3. Cryptanalysis of Shuai et al.'s Scheme

In this section, we demonstrate that Shuai et al.'s scheme suffers from various attacks when assuming the adversary armed with real-world capabilities [9–11] as below:

- (1) Exhaust all the items in the Descartes space of passwords and identities
- (2) Get  $ID_i$  when assess the security of the scheme

- (3) Intercept, eavesdrop, or resend the messages in the open channel
- (4) Get the data stored in the smart device
- (5) Get previous session keys
- (6) Get the secret key  $K$  when accessing forward secrecy
- (7) The adversary can be the administrator of the registration authority

*3.1. Offline Dictionary Attack.* When the adversary gets the data  $\{A_1, A_2, A_3\}$  stored in the victim  $U_i$ 's mobile device, she can guess  $U_i$ 's password and identity correctly as the following steps:

The attack steps are as follows:

- Step 1.* Guess  $PW_i$  to be  $PW_i^*$ ,  $ID_i$  to be  $ID_i^*$ .
- Step 2.* Compute  $a^* = A_2 \oplus h(ID_i^* || PW_i^*)$ .
- Step 3.* Compute  $HPW_i^* = h(PW_i^* || a^*)$ .
- Step 4.* Compute  $A_3^* = h(ID_i^* || HPW_i^*)$ .
- Step 5.* Verify the correctness of  $PW_i$  and  $ID_i$  by checking if  $A_3^* == A_3$ .
- Step 6.* Repeat Steps 1–5 until the equation holds.

The time complexity is  $O(|D_{PW}| * |D_{id}| * 3T_H)$ , where  $T_H$  is the time of the hash function.

Assuming the adversary gets the victim's identity  $ID_i$ , the adversary, with the data stored in the smart device and transmitted in the open channel, can guess  $U_i$ 's password successfully as below:

The attack steps are as follows:

- Step 1.* Guess  $PW_i$  to be  $PW_i^*$ ,  $ID_i$  to be  $ID_i^*$ .
- Step 2.* Compute  $a^* = A_2 \oplus h(ID_i^* || PW_i^*)$ .
- Step 3.* Compute  $HPW_i^* = h(PW_i^* || a^*)$ .
- Step 4.* Compute  $K_{GU}^* = A_1 \oplus HPW_i^*$ .
- Step 5.* Compute  $R_1^* || SID_k = M_1 \oplus K_{GU}^*$ .
- Step 6.* Compute  $V_1^* = h(ID_i || R_1^* || K_{GU}^* || M_1)$ .
- Step 7.* Verify the correctness of  $PW_i$  and  $ID_i$  by checking if  $V_1^* == V_1$ .
- Step 8.* Repeat Steps 1–6 until the correct value of  $PW_i$  is found.

The time complexity is  $O(|D_{pw}| * |D_{id}| * 3T_H)$ .

Possible Countermeasures: In offline dictionary attack, the inherent causes are as follows: (1) the adversary can find

a verifier to check the correctness of the guessed password and (2) to the adversary, the verifier only contains one unknown parameter (i.e., the victim's password), that is, all the parameters which consist of the verifier can be derived from the victim's password. According to Wang and Xu [12], the offline dictionary attack can be divided into two types in terms of where the verifier is from. In the former attack, the verifier  $A_3$  is extracted from the smart device. To deal with this attack, Wang and Wang [13] proposed a way of integrating the fuzzy-verifier technique and honeywords. That is, let  $A_3 = h(ID_i || HPW_i) \bmod n_0$ , where  $n_0$  is an integer and  $2^4 \leq n_0 \leq 2^8$ .

As such, there are about  $|D_{id} * D_{pw}| / l_0 \approx 2^{32}$  candidate pairs of identity and password which satisfy the equation of Step 5, when  $l_0 = 2^8$ . To test the specific pair of identity and password, the adversary needs to initiate the access request online, and this (the failure attempt) can be detected and stopped by the parameter TEMP.

To the second attack, a public key is necessary [14]. In Shuai et al.'s scheme, we need to set the verifier  $V_i = h(ID_i || R_1 || K_{GU} || M_1 || A_5)$  and  $DID_i = ID_i \oplus h(A)$ . As such, there are essentially two unknown parameters to the adversary, i.e., the password and  $A_5$ , and the space of  $A_5$  is too large for the adversary to conduct the offline dictionary attack.

*3.2. Forward Secrecy.* Forward secrecy requires that the exposure of the secrecy key  $K$  will not affect the security of previous conversations. However, we find this scheme cannot provide forward secrecy. If the adversary gets  $K$  and eavesdrops the parameters  $\{M_2, M_3\}$ , she can get the session key SK as the following steps:

The attack steps are as follows:

- Step 1.* Compute  $K_{GS}^* = h(SID_k || K)$ .
- Step 2.* Compute  $(ID_i^* || GID_j^* || R_1^* || R_2^*) = M_2 \oplus K_{GS}^*$ .
- Step 3.* Compute  $R_3^* = M_3 \oplus K_{GS}^*$ .
- Step 4.* Compute  $SK = h(ID_i^* || GID_j^* || R_1^* || R_2^* || R_3^*)$ .

The time complexity is  $O(|D_{pw}| * |D_{id}| * 2T_H)$ .

Possible Countermeasures: According to Ma et al. [14], the public key technique and two modular exponentiation or point multiplication operations on the smart device are required. Following this principle, we can let  $SK = h(ID_i || GID_j || A_4 || A_6 || A_7)$ , where  $A_6 = R_3 \cdot P$ ,  $A_7 = \omega \cdot A_6 = R_3 \cdot A_4 \cdot A_6$  is computed by  $SD_k$  and should be transmitted to  $U_i$  in the open channel.  $A_4$  also needs to be sent to  $SD_k$ .  $R_3$  cannot be transmitted to any participants. As such, the adversary has no way to compute  $A_7$  (it is a computational difficult problem which cannot be solved within polynomial time), and the forward secrecy is achieved.

*3.3. Inside Attack.* Suppose the adversary is also the administrator of RA, then she can exploit the register message and the data stored in mobile devices to guess the victim's password as follows:



The attack steps are as follows:

*Step 1.* Guess  $PW_i$  to be  $PW_i^*$ ,  $ID_i$  to be  $ID_i^*$ .

*Step 2.* Compute  $a^* = A_2 \oplus h(ID_i^* || PW_i^*)$ .

*Step 3.* Compute  $HPW_i^* = h(PW_i^* || a)$ .

*Step 4.* Verify the correctness of  $PW_i$  and  $ID_i$  by checking if  $HPW_i^* == HPW_i$ .

*Step 5.* Repeat Steps 1–4 until the correct value of  $PW_i$  and  $ID_i$  is found.

The time complexity is  $O(|D_{pw}|^* |D_{id}|^* 2T_H)$ .

Possible Countermeasures: Inside attack is practical although it has high requirements on the adversary's capability. In this scheme, the verifier  $HPW_i$  contains  $PW_i$  and  $a$ , and  $a$  can be computed using the parameters in the mobile device. Therefore, a way to deal with this attack is to update  $a$  after the registration. After receiving the response from RA, the user side should select a new random nonce  $a'$ , update  $HPW_i$  as  $h(PW_i || a')$ , and then set  $A_2 = a' \oplus h(ID_i || PW_i)$  and  $A_3 = h(ID_i || HPW_i)$ .

#### 4. Conclusion

In this paper, we have analyzed an anonymous authentication scheme for a smart home environment proposed by Shuai et al. [8]. We demonstrated that their scheme suffers from various attacks although it is proved to be secure under the random oracle model. We showed that this scheme cannot resist offline dictionary attack and inside attack and also fails to provide forward secrecy. After pointing out these security flaws, we proposed possible countermeasures to deal with them. These suggestions can also be applied to most similar schemes. Thus, our work is helpful to the design of a secure and efficient user authentication scheme for the smart home environment.

#### Data Availability

No data were used to support this study.

#### Conflicts of Interest

The authors declare that they have no conflicts of interest.

#### References

- [1] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K. K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9390–9401, 2020.
- [2] X. Huang, Y. Xiang, A. Chonka, J. Zhou, and R. H. Deng, "A generic framework for three-factor authentication: preserving security and privacy in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1390–1397, 2010.
- [3] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.
- [4] F. Wang, G. Xu, G. Xu, Y. Wang, and J. Peng, "A robust IoT-based three-factor authentication scheme for cloud computing resistant to session key exposure," *Wireless Communications and Mobile Computing*, vol. 2020, 15 pages, 2020.
- [5] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ECC-based provable secure authentication protocol with privacy preserving for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3599–3609, 2018.
- [6] F. Wang, G. Xu, and G. Lize, "A secure and efficient ECC-based anonymous authentication protocol," *Security and Communication Networks*, vol. 2019, 13 pages, 2019.
- [7] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.
- [8] M. Shuai, N. Yu, H. Wang, and L. Xiong, "Anonymous authentication scheme for smart home environment with provable security," *Computers & Security*, vol. 86, no. 2019, pp. 132–146, 2019.
- [9] Q. Jiang, J. Ma, F. Wei, Y. Tian, J. Shen, and Y. Yang, "An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 76, pp. 37–48, 2016.
- [10] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.
- [11] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.
- [12] C. Wang and G. Xu, "Cryptanalysis of three password-based remote user authentication schemes with non-tamper-resistant smart card," *Security and Communication Networks*, vol. 2017, 14 pages, 2017.
- [13] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2018.
- [14] C. Ma, D. Wang, and S. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *International Journal of Communication Systems*, vol. 27, pp. 2215–2227, 2012.

## Research Article

# From Hardware to Operating System: A Static Measurement Method of Android System Based on TrustZone

Xinhong Hei, Wen Gao , Yichuan Wang , Lei Zhu, and Wenjiang Ji

*School of Computer Science and Technology, Xi'an University of Technology, Xi'an, China*

Correspondence should be addressed to Yichuan Wang; [chuan@xaut.edu.cn](mailto:chuan@xaut.edu.cn)

Received 21 April 2020; Revised 27 July 2020; Accepted 7 September 2020; Published 21 September 2020

Academic Editor: Qi Jiang

Copyright © 2020 Xinhong Hei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Android system has been one of the main targets of hacker attacks for a long time. At present, it is faced with security risks such as privilege escalation attacks, image tampering, and malicious programs. In view of the above risks, the current detection of the application layer can no longer guarantee the security of the Android system. The security of mobile terminals needs to be fully protected from the bottom to the top, and the consistency test of the hardware system is realized from the hardware layer of the terminal. However, there is not a complete set of security measures to ensure the reliability and integrity of the Android system at present. Therefore, from the perspective of trusted computing, this paper proposes and implements a trusted static measurement method of the Android system based on TrustZone to protect the integrity of the system layer and provide a trusted underlying environment for the detection of the Android application layer. This paper analyzes from two aspects of security and efficiency. The experimental results show that this method can detect the Android system layer privilege escalation attack and discover the rootkit that breaks the integrity of the Android kernel in time during the startup process, and the performance loss of this method is within the acceptable range.

## 1. Introduction

In recent years, with the rapid development of mobile Internet technology, the number of users using Android mobile devices has increased rapidly. By 2018, the share of the Android system in the global smartphone has reached 85% [1]. According to CVE details [2], in 2017 and 2016, the vulnerability of the Android operating system was 842 and 523, respectively. According to the classification of these vulnerabilities in literature [3], the ratio of kernel vulnerabilities and standard libraries is the largest, accounting for 41% and 32%, respectively. At present, the Android system is mainly faced with cross script attack, privilege promotion attack, malware attack, privacy stealing attack, replay attack, communication attack, NFC attack, denial of service attack, etc. [4–9]. However, for the protection of attacks, most of the current research is in the application layer [10–16], but these solutions cannot fundamentally solve the security problems encountered by the current mobile terminal, and the terminal may still be threatened by malicious attackers and malware, so we should start from the system layer of the

mobile intelligent terminal and build a secure and reliable mobile terminal system from bottom to top to ensure the security of intelligent terminal.

At present, there are three main methods for the security research of the Android system layer: SEAndroid, hardware-assisted virtualization technology, and TrustZone technology based on ARM. The introduction of SEAndroid has largely prevented malicious applications from attacking the system, but SEAndroid needs to rely on a trusted kernel and cannot defend against direct attacks from enemies [17]. For the hardware virtualization technology, L4Android [18] adopts the hardware virtualization technology to isolate the Android system on each occasion, but the attack on the system cannot be stopped. The Droid Visor [19] protects the integrity of the static key objects of the kernel and detects the rootkits of processes and modules, but it cannot detect the rootkits that modify the dynamic entropy pool resources. [20] can detect the integrity of the Android system kernel, but it cannot defend against the rights raising attack. For TrustZone technology, Zhang et al. proposed that T-Mac used TrustZone technology to strengthen Mac [21] but did not consider other

factors affecting kernel security, such as not measuring the control flow in the kernel. Ahmed proposed a real-time kernel protection mechanism based on the advantage of TrustZone's hardware isolation. Although it has achieved some results against kernel level attacks, it has made significant modifications to the kernel. Ge et al. proposed a core code integrity measurement architecture SPROBES [22] based on the TrustZone architecture. Although it can measure rootkit, the performance loss of single instruction measurement is large. [23, 24] can implement a side-channel cache attack on the Android operating system using TrustZone. Therefore, in order to solve the security problem of the Android system layer, there is an urgent need for a more reliable and secure solution. Because SEAndroid needs to rely on a trusted kernel, hardware virtualization technology is currently considered too expensive and low versatility [25]. Therefore, this paper uses TrustZone technology to study the Android system layer kernel.

From the perspective of trusted computing, this paper proposes and implements a trust static measurement method for the Android system based on TrustZone, which takes bl1.bin image in ARM trusted firmware (ATF) as the trusted root, combines TrustZone technology with the Android system, and measures the kernel modules and executable files in the system startup process statically, and finally, extends the trusted root to the Android system application framework layer that provides a reliable underlying environment for the detection of the Android system application layer. This method can detect the elevated privilege attack of the Android system layer and discover the rootkit that breaks the integrity of the Android kernel in time during the startup process, and the performance loss of this method is within the acceptable range.

To sum up, our main contributions are as follows:

- (1) Using the idea of trusted computing, according to the MTM specification, the hardware device is regarded as the source of trust, and the trust chain for Android system startup is designed to solve the problem of trust from the source
- (2) A static measurement method for the Android operating system kernel is designed, which transfers the trust of the trusted root to the Android application framework layer through the trust chain

The rest of the paper is arranged as follows. The second section introduces the related knowledge of the technology used in this paper. The third section introduces our overall design. After that, the fourth section introduces the implementation process and gives the evaluation results in the fifth section. Finally, in the sixth section, we summarize this paper and look forward to the future work.

## 2. Related Work

*2.1. Android Trust Chain.* Since the establishment of the trusted computing organization (TCG), trusted computing has made rapid development. The establishment and transmission of the trust chain are the basic problems of trusted

computing, which involve three points: trust root, trust transmission, and trust measurement. Trust root is the cornerstone of system trust and also the starting point of trust transmission. Trust transmission refers to the function of providing complete trust to the upper layer. The implementation of each layer of the system is based on the trust of the next layer, and the extension of the system's trusted range can be realized through trusted transmission [26, 27]. Trusted measurement refers to the integrity verification of files and their related configuration information to prevent them from being tampered with. The trust chain constructed by these three points gives trust from bottom to top and reduces the trust management of a large-scale system to the root of trust.

*2.2. Android Framework Layer.* As the middle layer of the application layer and underlying code, the Android framework layer encapsulates standardized modules to provide Java API for the application layer and also includes the JNI method to call underlying library functions to provide some system services; for example, Cameraservice and MediaPlayerService are closely related to the user's privacy data. In the /system/framework directory of the Android system, there are mainly three types of files: jar package, ODEX file, and boot.art and boot.oat. Jar package provides support for various libraries in the framework layer for some functions of Android; for example, when executing the AM command, the am.jar file will be loaded. From Android version 4.4, Google has migrated the ART virtual machine to Android. After version 5.0, the ART virtual machine completely replaces the original Dalvik virtual machine. To run ART, the boot.art and boot.oat files in the directory are required. When compiling the Android source code, some common classes will be packaged into boot.oat; boot.art contains the pointer to the method code in boot.oat, which is the boot image of the ART virtual machine. The ODEX file in system/framework/oat/arm directory is the result of optimizing some jar packages when compiling source code. For example, services.odex will be loaded when creating system services.

Our goal is to measure the complete Android framework layer, so all files in the /system/framework directory are our goal.

*2.3. Selection of Experimental Technology.* At present, there are three main methods for the security research of the Android system layer: SEAndroid, virtualization technology, and TrustZone technology based on ARM. Because SEAndroid relies on a trusted kernel and cannot guarantee the security of the underlying system, we will not discuss it in this part. Therefore, we compare virtualization technology with TrustZone technology. The comparison results are shown in Table 1.

*2.3.1. Security.* All code resources in the trusted execution environment (TEE) are protected, and the management of this code requires certain permissions based on hardware control. The downloading and installation of trusted applications are also based on a certain trust. Particularly for trusted applications developed by third parties, the source of the

TABLE 1: Contrast result.

	TrustZone	Virtualization technology
Safety	Higher	Lower
SOC implementation	Easily	Difficulty
Ecology	Universal	No standardization
Application scenario	Sensitive applications	Applications that need to improve efficiency

application must be identified and certified before the application is downloaded and installed, so as to reduce malicious software, the attack of Trojan program on a safe operating system.

Compared with the security function of the TEE, virtualization technology allows multiple operating systems to execute on a host processor. Although these operating systems are isolated from each other, they do not make these operating systems have security features. Virtualization does not provide the corresponding interface to deal with security functions, let alone separate security hardware. From the perspective of isolating operating systems from each other to ensure the security of some operating systems, virtualization technology highlights the weakening.

**2.3.2. SOC Implementation.** For the SOC system, the TEE has the ability to control all hardware peripherals and filter the access to these peripherals under different CPU states, so the system itself needs to be clear about which execution environment is currently accessing which resources. For virtualization technology, the controller is only a software component, which can be directly connected to peripheral devices. The system itself does not perceive virtual machines. Because virtualization is only used to organize software running on the ARM core, it is very difficult to build a complete security system relying on it.

**2.3.3. Ecological Creation and Maintenance.** At present, the TEE has been deployed in a large number, and the platform it depends on can be completely transparent, and the TEE has the operating system agnostic. No matter what operating system is used by the mobile platform, it will have a set of standard communication interface to ensure that the operating system and the trusted application running in the TEE communicate with each other. On the contrary, virtualization products on mobile platforms do not have a standardized ecosystem to focus on the security needs of the industry. In addition, virtualization will be more intrusive at the following two levels: one is the virtual machine level, and the other is that the controller driver needs to adapt to each new platform monitor version.

**2.3.4. Application Scenario.** The TEE is generally used to implement sensitive applications, such as DRM, mobile financial payment, and enterprise mobile office. Virtualization technology enables multiple software environments to run on shared physical resources, so its use scenarios are more suitable for those application scenarios that improve efficiency.

In conclusion, TrustZone technology can better achieve the trusted static measurement of the Android system in this experiment.

**2.4. TrustZone and OP-TEE.** TrustZone is a group of hardware security extensions for ARM. The TrustZone space controller can divide DRAM into different memory areas and specify the memory area as safe or normal. The world executed by the processor is represented by an ns bit, which propagates through the system bus. The trusted bus structure ensures that normal world components cannot access any secure world resources [28]. The Open-source Portable Trusted Execution Environment (OP-TEE) project is implemented by the TEE open source launched by Linaro, which fully complies with the specifications and standards issued by the GP organization for TEE and supports all APIs of document specifications such as TEE client API v1.0 [29].

Therefore, this paper chooses a secure world `os(optee_os)` in OP-TEE as a trusted execution environment.

**2.5. File Encryption Key of OP-TEE.** FEK is the file encryption key used by OP-TEE when encrypting data. Each secure file of trusted application has a FEK to encrypt the data of the corresponding file. The generation process is shown in Figure 1.

**Secure storage key (SSK):** the value of the secure storage key is different in different devices. After the OP-TEE is started, the chip ID and hardware unique key (HUK) will be used to calculate the value through HMAC for use when generating other keys.

**Storage Trusted Storage Key (TSK):** TSK is the key used to generate file encryption key (FEK). TSK is calculated by HMAC using SSK as the key to the UUID of trusted application. TSK will be used to generate FEK finally.

The generation process of FEK is as follows:

$$\begin{aligned}
 \text{SSK} &= \text{HMAC}(\text{HUK}, \text{message}), \\
 \text{Message} &:= \text{concatenate}(\text{chip\_id}, \text{string\_for\_ssk\_gen}), \\
 \text{TSK} &= \text{HMAC}(\text{SSK}, \text{TA\_UUID}), \\
 \text{FEK} &= \text{AES\_CBC}(\text{TSK}, \text{in\_key}),
 \end{aligned} \tag{1}$$

where `in_key` is the random number needed to generate FEK.

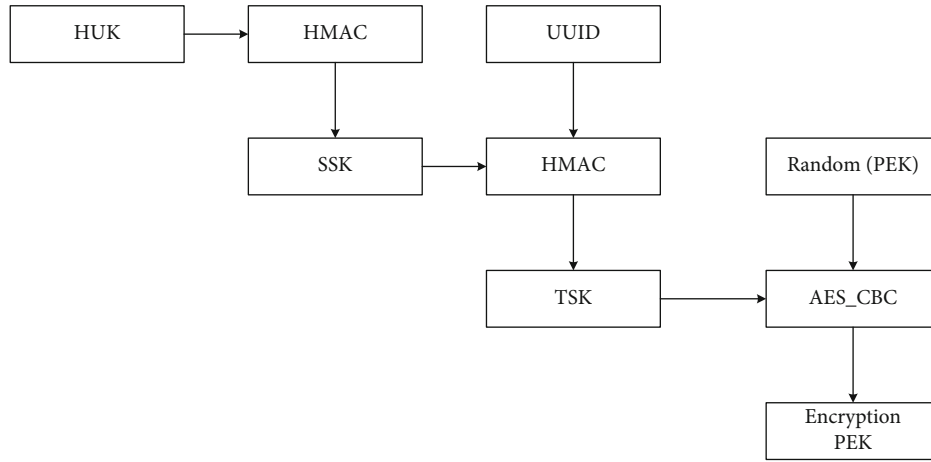


FIGURE 1: File encryption key (FEK) generation process.

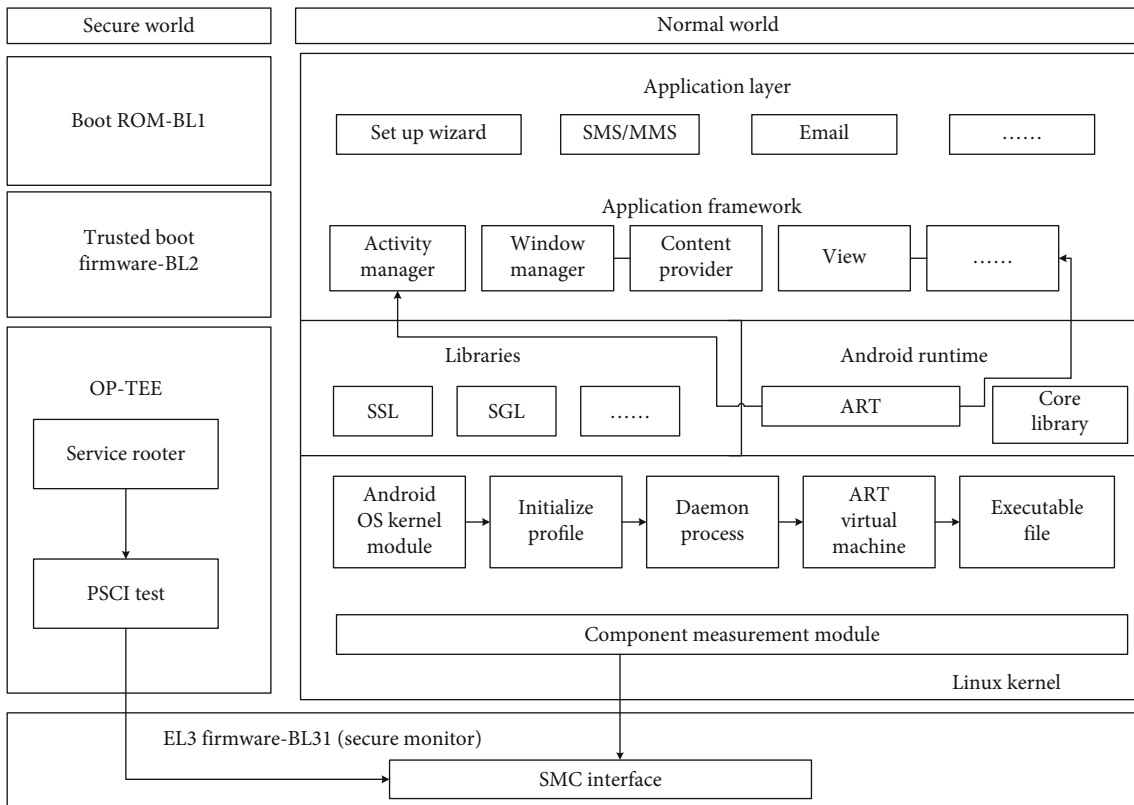


FIGURE 2: System architecture design.

### 3. System Design

In this part, we first introduce the design of the system architecture, then describe the trusted verification process and the trusted static measurement method in detail during the startup of the Android system.

*3.1. Architecture Overview.* According to the MTM standard of a trusted system, to establish the whole system’s trust, we need to establish a trusted root first, then form a trusted chain according to the detection, and transfer the trust to each module of the system. In order to achieve the trusted static

measurement in the process of Android system startup, we combine ARM trusted firmware with OP-TEE which implements TrustZone technology with the Android system, take bl1.bin image as the trusted root and OP-TEE as the trusted storage root, and add the degree module in the Android system kernel layer to design a trusted static measurement method for the Android system. The overall framework of the system is as shown in Figure 2.

*3.2. Trust Delivery Process.* According to the architecture chart we designed, the flow of the system integrity verification mechanism we designed is bl1 → bl2 → bl31 → optee\_

os  $\rightarrow$  Bootloader  $\rightarrow$  kernel  $\rightarrow$  Android system  $\rightarrow$  APP. The whole process of trusted authentication is shown in Figure 3.

The startup process is divided into trusted execution environment (TEE) side startup and Rich Execution Environment (REE) side startup, which are described in the following two aspects.

**3.2.1. TEE Side Start Process.** After the system is powered on, it will start to execute the code in chip ROM. Chip ROM will first jump to the bl1.bin image of ATF for execution. After bl1 completes the operation of loading the bl2.bin image into RAM and setting the interrupt vector table, it will perform the signature verification operation on the bl2 image file. During the compilation of ATF, the system will perform the SHA256 calculation on all levels of images in ATF and then sign the generated summary. The private key is the RSA2048 key under the directory file. If the verification is passed, call the EL3, exit function to realize the jump from bl1 to bl2, and enter bl2 to start execution. In bl2, the signature verification module of the image file will be initialized first.

If the signature verification passes, the image file of the bootloader of bl31, OP-TEE, and Android system will be loaded into the memory with corresponding permission. Among them, bl31 is the execution software of EL3, whose function is to call security monitoring mode (SMC) instructions and interrupt processing. After triggering the security monitoring mode call in bl2, bl31 starts to run. bl31 determines whether to load OP-TEE by parsing whether there is an entry function of OP-TEE and verifying the validity of the OP-TEE image signature. If the entry function exists and the image signature verification is passed, OP-TEE will be started. After OP-TEE, the security monitoring mode call will be triggered to reenter bl31 for further execution. bl31 obtains the bootloader image file of the next Android system that needs to be loaded into the Rich Execution Environment (REE) side by querying the link list and verifies the validity of the bootloader file. If the verification is passed, then set the CPU state and running environment when the REE side is running and exit EL3 to enter the bootloader image startup of the Android system. At this time, the trust of the trusted root is transferred from bl1 to the bootloader of the Android system. If any part of the above process fails to be verified, it will directly cause the system to hang up.

**3.2.2. REE Side Start Process.** When the bootloader starts to start, it enters into the normal world of Android system startup. In the startup of the REE side, as shown in the architecture design in Figure 1, we add a measurement module to the kernel layer of the Android system. In order to formally describe and verify the startup process of the REE side, we refer to the PKI trust model on the basis of reference [30] and first give the following definitions:

*Definition 1.* Let  $e^*$  be the set of all components involved in the safe startup, and  $m$  be the OP-TEE,  $\forall c_i, c_j \in e^*$ , where  $i, j \in N$ . The following are the propositions:

- (1) Integrity measurement capability:  $\text{TrustCapa}(c_i, c_j, \text{Integ}) | \langle p_{jm} \rangle$ . It indicates that when the constraint condition  $p_{jm}$  is satisfied,

component  $c_i$  believes that  $c_j$  has the trusted integrity measurement capability;  $p_{jm}$  refers to the trusted measurement capability that component  $c_j$  can communicate with OP-TEE

- (2) Integrity credibility:  $\text{Trusted}(c_i, c_j, \text{Integ})$  indicates that component  $c_i$  believes that  $c_j$  has a trusted integrity measurement attribute
- (3) Integrity measurement:  $\text{Meas}(c_i, c_j, \text{Integ}) | \langle \text{RIM} \rangle$  indicates that component  $c_i$  measures the integrity value of  $c_j$ , which is the same as the reference integrity value (RIM) stored in OP-TEE

*Definition 2.* Let  $e^*$  be the set of all components involved in the safe startup, and  $m$  be the OP-TEE,  $\forall c_i, c_j, c_k \in e^*$ . The following are the propositions:

- Rule 1. Integrity measurement capability transfer rule:-  
 $\text{TrustCapa}(c_i, c_j, \text{Integ}) | \langle p_{jm} \rangle \wedge \text{TrustCapa}(c_j, c_k, \text{Integ}) | \langle p_{km} \rangle \rightarrow \text{TrustCapa}(c_i, c_k, \text{Integ}) | \langle p_{km} \rangle$   
 Rule 2. Trust delivery rule:-  
 $\text{TrustCapa}(c_i, c_j, \text{Integ}) | \langle p_{jm} \rangle \wedge \text{Meas}(c_i, c_j, \text{Integ}) | \langle \text{RIM} \rangle \rightarrow \text{Trusted}(c_i, c_k, \text{Integ})$

At the start of trusted start, the external observer  $c_0$  thinks that only bootloader ( $c_1$ ) in the mobile intelligent terminal is trusted and has integrity measurement capability, so there are initialization conditions as follows:

$$\begin{aligned} & \text{Trusted}(c_0, c_1, \text{Integ}), \\ & \text{TrustCapa}(c_i, c_j, \text{Integ}) | \langle p_{1m} \rangle. \end{aligned} \quad (2)$$

Android kernel integrity measurement module is responsible for measuring the kernel module loaded in Android intelligent mobile terminal, initialization configuration file, daemons, ART virtual machine initialization process, and all executable files under the framework layer. The measurement process is as follows.

The measurement module measures the Android OS kernel module ( $c_2$ ) and compares the measurement value with the expected measurement value stored in OP-TEE. If the measurement result is consistent, the next measurement will be continued. At this time,

$$\text{TrustCapa}(c_0, c_1, \text{Integ}) | \langle p_{1m} \rangle \wedge \text{Meas}(c_1, c_2, \text{Integ}) | \langle \text{RIM} \rangle \rightarrow \text{Trusted}(c_0, c_2, \text{Integ}). \quad (3)$$

Since  $c_2$  has started and initialized  $m$ , it can be seen from the assumption that

$$\text{Meas}(c_1, c_2, \text{Integ}) | \langle \text{RIMCert} \rangle \rightarrow \text{TrustCapa}(c_1, c_2, \text{Integ}) | \langle p_{2m} \rangle. \quad (4)$$

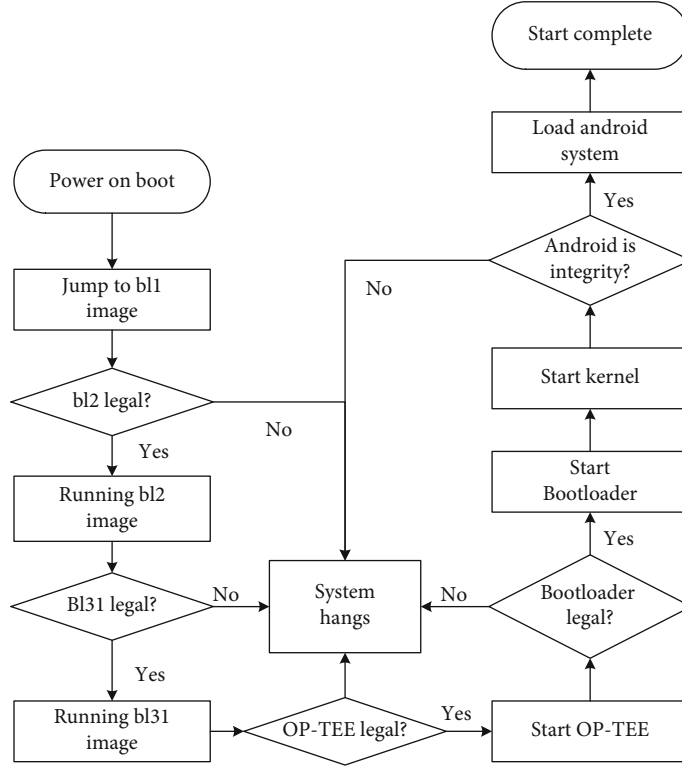


FIGURE 3: Start process trusted authentication process.

The measurement module measures the Android initialization configuration file ( $c_3$ ) and compares the measurement value with the expected measurement value stored safely in OP-TEE. If the measurement result is consistent, continue to the next measurement, and the same can be obtained from the above derivation. At this time,

$$\text{Trusted}(c_0, c_3, \text{Integ}), \text{TrustCapa}(c_2, c_3, \text{Integ}) | \langle p_{3m} \rangle . \quad (5)$$

According to the above method, the daemons ( $c_4$ ) are measured and the measurement values are verified. If the results are consistent, then

$$\text{Trusted}(c_0, c_4, \text{Integ}), \text{TrustCapa}(c_3, c_4, \text{Integ}) | \langle p_{4m} \rangle . \quad (6)$$

The measurement module measures the initialization process of the ART virtual machine ( $c_5$ ) and verifies the measurement value. If the result is consistent, then

$$\text{Trusted}(c_0, c_5, \text{Integ}), \text{TrustCapa}(c_4, c_5, \text{Integ}) | \langle p_{5m} \rangle . \quad (7)$$

Finally, measure and verify all executable files under the framework layer of the Android system. If the results are con-

sistent, then

$$\text{Trusted}(c_0, c_6, \text{Integ}), \text{TrustCapa}(c_5, c_6, \text{Integ}) | \langle p_{6m} \rangle . \quad (8)$$

It can be seen from the derivation that in the process of building the trusted start on the REE side, the trust relationship extends from  $c_1$  to the boundary  $c_6$  of the trusted base, indicating that the components on the trust chain in the trusted base are all trusted under the premise that the constraints are met. Therefore, the following conclusions can be drawn:

$$\text{Trusted}(c_0, c_i, \text{Integ}), \forall c_i \in e^* \quad 1 \leq i \leq 6 \quad i \in N. \quad (9)$$

By using the initial conditions and the formal deduction of the above formula, it can be seen that the safe startup process on the REE side is safe and reliable, which meets the requirements of integrity and trust verification. At this point, the trusted startup process of the whole system is completed, and the trust is extended from the root of trust for measurement to the framework layer of the Android system.

#### 4. Detailed Description of Scheme

In order to realize the architecture we designed in the previous section, this part describes the process of our specific implementation architecture from the aspects of environment construction, trusted image production, image integrity

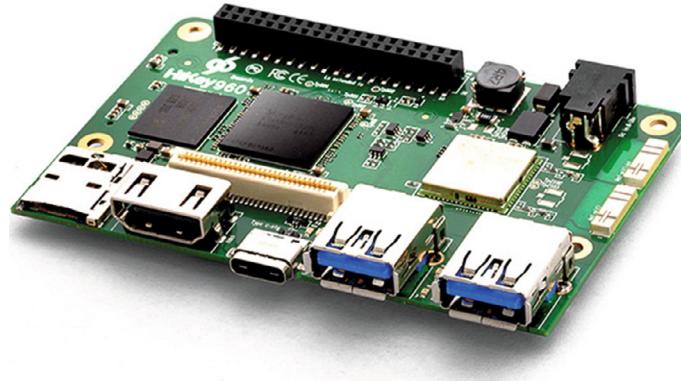


FIGURE 4: Kirin 960 SOC.

verification, measurement methods, and storage of expected metrics.

In this paper, in terms of the experimental hardware, the Huawei Kirin hikey960 development board based on Kirin 960 SOC shown in Figure 4 is used. The experimental environment is the Ubuntu 14.04 system.

**4.1. Environment Building.** First, get the latest Android AOSP and OP-TEE code of Google and then carry out MD5 detection and compare it with the official MD5 value to ensure the purity of the code. Then, add the TEE supplicant service in the `init.common.rc` file of Android source code and add the `optee-packages.mk` configuration file in the `linaro/hikey` directory. Add the configuration of OP-TEE in the `device-common.mk` configuration file and modify `conf.mk` and `platform_config.h` files of OP-TEE source code. The purpose is to identify and call the services provided by OP-TEE and provide a trusted environment for the next trusted measurement and safe storage in the Android system startup process. Then, obtain the source code of the underlying firmware ATF officially provided by ARM. The source code of ATF is divided into five parts: `bl1`, `bl2`, `bl31`, `bl32`, and `bl33`. `bl1`, `bl2`, and `bl31` are fixed firmware; `bl31` will execute the runtime service init function, which will call the initialization functions registered to all services in EL3. One of them is the TEE service. After the service is initialized, we modify the `bl32` init code in `bl31` to make the `bl32` executed function jump to OP-TEE and start the startup of OP-TEE. After the initialization of OP-TEE, `bl31` finds the bootloader of Android that needs to be executed by obtaining the link list of `bl2`, exits EL3, and enters the bootloader image for execution.

**4.2. Production of Trusted Image.** According to the Android system startup process framework described in the previous section, ARM trusted firmware, as a newly added stage of the secure startup architecture, not only completes the functions similar to some boot loader functions but also includes the module to verify the image in the next stage and the decryption public key at the time of verification. In order to realize the startup image integrity authentication, we recreate the `bl2`, `bl31`, OP-TEE, and bootloader images to be detected and make the trusted startup integrity verification image as shown in Figure 5.

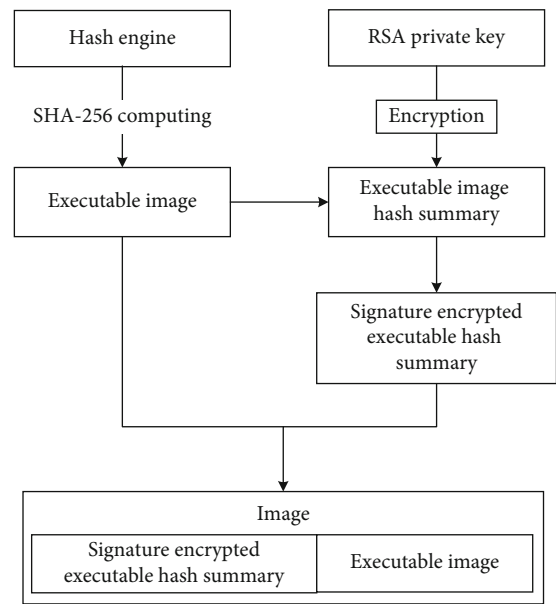


FIGURE 5: Production of the integrity verification image.

The steps are as follows: first, prepare the source code transplanted in each stage according to the requirements and compile and generate the executable image file; in the local computer, hash the executable image with the hash engine, which uses the public hash algorithm SHA-256. Get the hash result corresponding to the executable image: the hash summary of the image; then, use the RSA private key provided by the trusted firmware to sign and encrypt the asymmetric algorithm of the hash summary of the image. The encryption algorithm adopts the RSA asymmetric public key encryption algorithm; get the result after signature encryption; finally, the hash summary after signature encryption is relinked with the original executable image to generate the final image file.

**4.3. Image Integrity Verification.** In order to achieve image integrity verification, it is necessary to verify the source and integrity of the image in the next stage. After power on and startup, the system performs integrity detection to ensure the safe and tamper-free behavior of the image at startup. Figure 6 shows the oververification process in each stage.



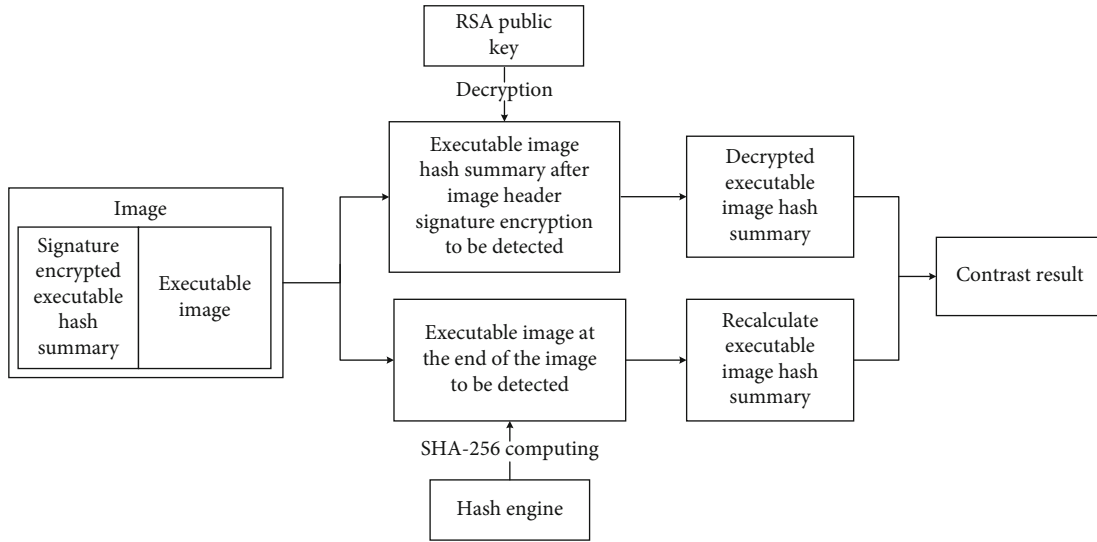


FIGURE 6: Image integrity verification diagram.

The verification process is as follows:

- (1) First, copy the image of the next stage to the designated memory location according to the requirements of the design startup process
- (2) The image is divided into two parts: one is the image head: the encrypted executable image hash summary; the other is the image tail: the executable image
- (3) The encrypted executable image hash digest is decrypted according to the public key stored in the executing domain
- (4) If it can be decrypted, it means that the image header data source is trusted, and the decrypted result can be obtained: the image hash summary can be executed, if it cannot be decrypted; it means that the image source is illegal and untrusted, and the operation of shutdown can be performed
- (5) Then, hash the executable image at the end of the image. The hash algorithm is a public hash algorithm and must be consistent with the algorithm adopted in the local image production to get the recalculated executable image hash summary
- (6) Compare the recalculated executable image hash summary with the result of the previous decryption operation
- (7) If the two hash values are the same, it means that the image is reliable and complete, and the verification is passed; if the two hash values are different, it means that the image is incomplete, and the shutdown operation is performed

**4.4. Implementation of the Measurement Method.** We transplant and modify IMA of the Linux kernel to realize kernel measurement during startup. The full name of IMA is integ-

rity measurement architecture; this component uses the hook function provided by LSM to detect files and application codes completely before they are executed or mapped to memory and generates a detection list. By reconstructing IMA code, we use the SHA-1 algorithm to measure kernel module, initialization configuration file, daemons, ART virtual machine initialization process, and executable files under the framework layer; the kernel is configured through making menuconfig; the kernel is recompiled; and the IMA service is started before the mount system partition.

**4.5. Storage of Expected Measure List.** Use the above measurement method to measure the kernel module, initialization configuration file, daemons, ART virtual machine initialization process, and the executable file of the framework layer of pure Android and generate the measurement list as the expected measurement value. Some expected measurement values generated are shown in Table 2.

Then, we store the generated measure list into the secure file system of the secure operating system as the expected metric list, which is used to start the comparison template for generating the metric list in the future. The security stored procedure steps are as follows:

- (1) The REE side initiates the encryption request, and the client CA that executes the `TEEC_InitializeContext` function initializes the context of the TEE
- (2) CA calls the `TEEC_OpenSession` function opens the session and establishes a connection with the corresponding trusted encryption and decryption program TA in the TEE
- (3) CA implements `TEEC_RegisterSharedMemory` registers a piece of shared memory for communication between CA and TA, which is used to transfer data and commands to the security service in the TEE and receive the results returned by the security

TABLE 2: Expected measure list.

/system/lib64/libjavacore.so	Sha1:825341bd045d62c15fd7bdc4ec026932ccff4178
/system/lib64/libopenjdk.so	Sha1:fc483a0156f5baf526bbcc9c90cd38b190516c89
/system/lib64/libvixl-arm.so	Sha1:8e6b911f86c4239a9bbd38df88fc1b91c5387f1d
/system/framework/core-oj.jar	Sha1:811d092eec40e1922af7aaf6189363de0f8a975f
/system/framework/core-libart.jar	Sha1:d2e8e403c1d0ddfecadc2c3ac51197f593e84dc0
/system/framework/okhttp.jar	Sha1:e26a028a129bd9c779667d03e2eedf9ea6ce6b7

service. If the memory allocation is successful, step (4) is executed; otherwise, step (6)

- (4) CA calls TEE\_CreatePersistentObject interface, TEE\_OpenPersistentObject interface, and TEE\_WriteObjectData function, respectively, and writes the data to be transferred into the registered shared memory. After receiving the command, the security service in the TEE first reads the data information in the shared memory, and then OP-TEE sends an RPC request to notify tee\_supplicant to complete the operation of the file system on the REE side and stores the security files in the data/TEE directory
- (5) Execute the TEEC\_ReleaseSharedMemory function to release shared memory
- (6) Execute the TEEC\_CloseSession function to close the session; the storage result is shown in Figure 7

**4.6. Secure Transfer of Measure List.** In the nonsecure environment, before the measure list file generated during the startup of the Android system on the REE side is transferred to the TEE security environment for comparison, the measurement data in this stage is also very easy to be intercepted by malicious programs. Therefore, we establish a secure metric list transmission channel between optee\_os and Android systems through the TrustZone driver module to ensure that the metric list is transmitted to optee\_os security. Figure 8 shows the framework of the security transmission channel of the measure list.

First of all, after generating the measurement list during the startup process of the Android system, REE obtains the key from the security environment and then encrypts the metric list with the aes-256 symmetric encryption algorithm. Then, it calls the CallTrustZone function through the TrustZone driver module to fall into the monitor environment. The monitor switches the execution environment of the system to the secure environment protected by TrustZone, the decryption module is called to decrypt the transmitted ciphertext, and then, the obtained metric list file is compared for the next operation.

**4.7. Comparison of Measure List.** The comparison phase is divided into two parts: first, decrypt the expected measure list file of the security storage, calling the read interface in TA and calling the syscall\_storage\_obj\_read function to read the data of the security file in the OP-TEE kernel space. The function first obtains the TA session ID, the running context and checks the permissions, and then calls the ree\_

fs\_read function to realize the operation of reading data. The second part is the comparison of measurement list files. SHA-1 operation on the decrypted expected measure list file is performed, and at the same time, SHA-1 operation is also performed on the measure list file decrypted in part 4.6. If the two results are consistent, the result will be returned to the REE side, and the Android system will start normally. If the results are different, a warning will pop up after the Android system starts.

## 5. Evaluation

In this part, we discuss the experimental results about the functional effectiveness and performance of our method. All experiments are carried out on the hikey 960 development board.

**5.1. Security Assessment.** In the five attacks, the first two modified several bytes of the syscall table subroutine to achieve the attack, the third one modifies the system's exception vector table, the fourth one injects malicious code into the trigger mechanism onTouchEvent() function to enhance the permissions of the kernel layer, and the fifth one removes the process from the list of processes in the kernel to hide the process. Reference [20] proposes an android kernel measurement method based on the ARM virtualization extension called DIMDroid. This experiment is compared with the static measurement method in DIMDroid, and the results are shown in Table 3.

From the measurement results, it can be seen that both tampering with kernel static measurement objects such as system call table and interrupt call table and process hiding can be detected. However, DIMDroid measurement cannot detect the privilege attack of the application framework layer and kernel layer.

In the measurement list storage process, we compare our secure storage scheme with the traditional scheme that measure list is stored in ordinary Android files. The results are shown in Table 4.

As the template resource of the Android system startup process measurement list, the expected measure list is the benchmark and basis of the whole comparison process. Because the list of expected measures is stored in a secure isolated area, it can block security threats from nonsecure environments. In addition, in order to prevent other security services in optee\_os from obtaining the expected measure list file, the asymmetric encryption algorithm combined with the key stored in the isolated area is used to complete the encryption protection of the expected measure list file.

```

M/TA: [WRITE] start to write file: wen.txt
D/TC:0 tee_ta_init_pseudo_ta_session:293 Lookup pseudo TA 59e4d3d3-0199-4f74-b94d-53d3daa57d73
D/TC:0 tee_ta_init_user_ta_session:637 Lookup user TA 59e4d3d3-0199-4f74-b94d-53d3daa57d73 (Secure Storage TA)
D/TC:0 tee_ta_init_user_ta_session:637 Lookup user TA 59e4d3d3-0199-4f74-b94d-53d3daa57d73 (REE)
D/TC:0 ta_load:317 ELF load address 0x103000
M/TA: Sec storage TA_CreateEntryPoint

```

FIGURE 7: Storage of expected measure list.

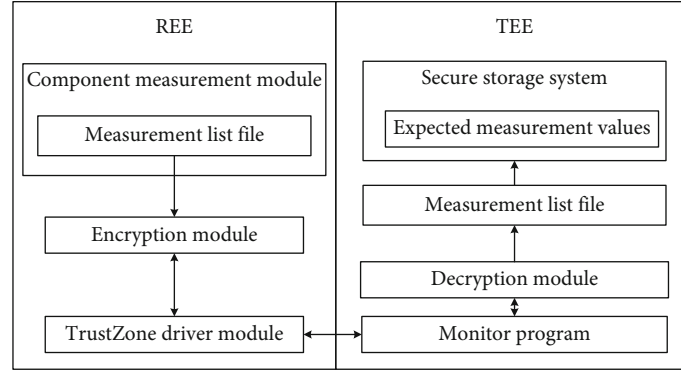


FIGURE 8: Secure transport framework.

TABLE 3: Attack experiment measurement results.

Rootkit	Attack function category	Measurement results of this experiment	DIMDroid metric
Rootkit1	Modify some bytes of syscall subroutine	√	√
Rootkit2	Modify some items of syscall	√	√
Rootkit3	Modify SWI software interrupt jump offset	√	√
Rootkit4	Inject malicious code into the onTouchEvent() function and elevate the kernel layer permissions to complete attack	√	×
Rootkit5	Intercept the proc_lookup function to hide the process	√	√

TABLE 4: Compare results.

Measure list	Our scheme	Traditional scheme
Storage location	Single file	optee_os
Safety	Weak	Strong
Encryption process	Unsafe	Safe

The attack of the Android system starting process metric list transmission process mainly occurs in the stage of transmitting the metric list from the Android environment to the TEE system. The TrustZone driver module will request memory space at the kernel layer and copy the list of metrics generated during startup. Since the list of measurements generated in the whole stage exists in ciphertext, the security of the process is guaranteed.

**5.2. Efficiency Evaluation.** In the experiment, we need to hash the image file and the file that the Android system needs to be measured. However, which hash algorithm to choose is our first consideration. Therefore, we choose four files with different sizes from the image file that need to hash and the file that the Android system needs to measure and do SHA-256,

SHA-1, and MD5 operations on them, respectively. The results are shown in Figure 9.

It can be seen from Figure 8 that with the increase of file size, SHA-256 has the longest calculation time and the largest growth rate for the file, while MD5 has the smallest overall calculation time and the least impact on the calculation rate by the file size. SHA-1 is between the two.

SHA-256, SHA-1, and MD5 are all unidirectional functions, which are almost irreversible. The information that will generate a complete summary is entered. However, it is possible for different information to generate the same summary, which is called a collision. The security of the hash function depends on the ability to resist strong conflict to a great extent. Therefore, to evaluate the security of the hash function, it is necessary to check whether the attacker can find a pair of conflicts under the existing conditions. Table 5 lists the conflict thresholds of three hash functions.

According to Table 5, SHA-256 has the highest security, while MD5 has the worst. Considering the above time and security results, for the hash operation of the image, since the number of images that need to be hash operation is four (bl2.bin, bl31.bin, op-tee os, and uboot.img), the number of images that need to be calculated is small, and we have high

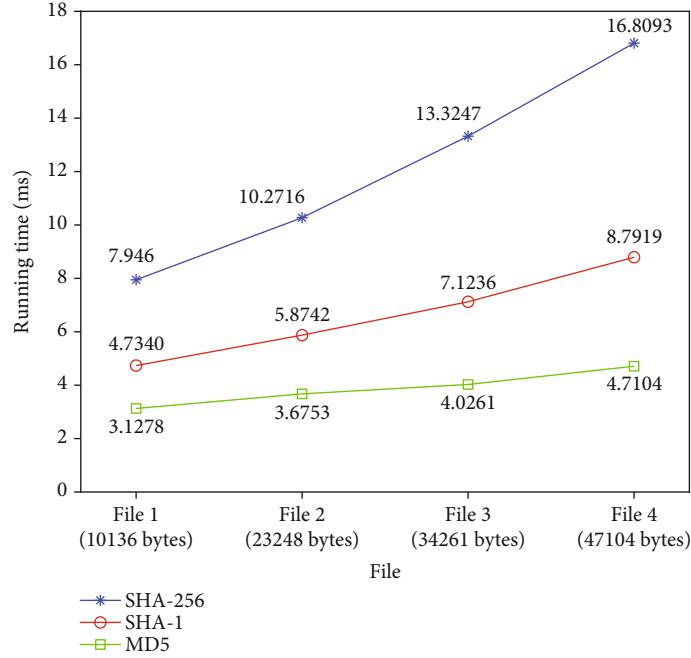


FIGURE 9: MD5, SHA-1, and SHA-256 encryption time.

TABLE 5: The collision thresholds for four commonly used hash functions.

Hash function	Function collision threshold
MD5	$2^{64} \approx 1.8 * 10^{19}$
SHA-1	$2^{80} \approx 1.2 * 10^{24}$
SHA-256	$2^{128} \approx 3.4 * 10^{38}$

security requirements for the image, so we choose the SHA-256 algorithm to generate a summary value for the image. For the measurement of Android system layer files, because the number of files to be measured is hundreds, if SHA-256 is selected, it will cause a lot of performance loss, so we choose SHA-1 operation to measure Android system layer files.

In this paper, for the scheme of trusted measurement of the Android system startup process, its performance impact mainly lies in the signature verification of image, the startup of OP-TEE, the SHA-1 operation on the set file, and the interaction time between the Android system and the OP-TEE. We have done 20 experiments on startup and take the average value of the results, as shown in Table 6.

In 20 experiments, the bootloader, kernel, and Android OS startup time is 10.2%, 8.3%, and 14.8% longer than that of the general Android. Because the startup of the trusted Android involves the time required for the startup of the trusted firmware and OP-TEE, compared with the normal Android startup process, the trusted startup process also increases the additional time overhead for the startup of ATF and OP-TEE. As shown in Figure 10, the starting time range of native Android is 21.1 s-22.8 s, and the starting time range of Android added to this experimental method is 26.3 s-27.6 s. The average starting time of this experiment is 23.4% longer than that of native Android.

TABLE 6: Time required to start components.

Unit (ms)	Bl1, Bl2, Bl31	OP-TEE	Bootloader	Kernel	Android OS
Normal start up	0	0	1226	5331	15052
Trusted startup	1219	2127	1352	5774	16484

In order to judge the impact of the kernel measurement module added on the REE side on the performance of the Android system, this paper uses the AnTuTu benchmark software, which is specialized in scoring Android device phones and tablets. Compared with the performance index of the unused kernel measurement module, the performance index mainly selects several mainstream options at this stage: ram speed, CPU floating-point calculation performance, and CPU integer calculation performance. Use the AnTuTu software test module to measure the kernel 100 times and take the average value. The performance loss ratio is the percentage of the difference between the score of the performance index item measured by the measurement module and the score of the index item measured by the measurement module, as shown in Table 7.

It can be seen from Table 6 that there is a certain performance loss in using the measurement module compared with not using the measurement module, but within the acceptable range, it shows that this method has certain reference significance for ensuring the integrity of the Android kernel.

## 6. Conclusion

In this paper, we propose and implement a TrustZone-based method to measure the trustworthiness of the Android

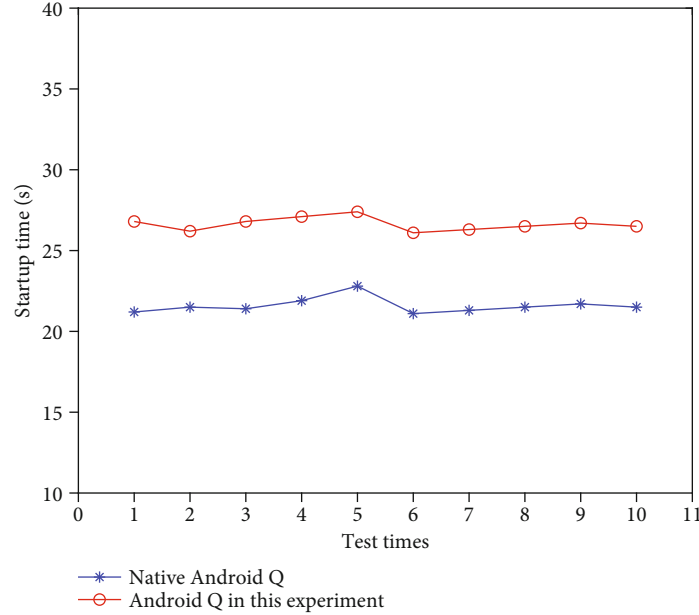


FIGURE 10: Startup time comparison.

TABLE 7: Using AnTuTu to test measurement module results.

Test item	Performance loss rate (%)
RAM speed	2.86
CPU floating-point calculation	2.58
CPU integer calculation	6.21

system. We use the bl1 image in ARM trusted firmware (ATF) as the trusted root, combine TrustZone technology with the Android system to measure the kernel modules and executable files in the system startup process, and finally, extend the trusted root to the entire Android platform. The next step is to give different weight values to different files according to the startup relationship, judge the security of the system according to the sum of the weight values, and give a more comprehensive and reasonable measurement verification to the Android system.

### Data Availability

The data used to support the findings of this study are included within the article.

### Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

### Acknowledgments

This research work is supposed by the National Key R&D Program of China (2018YFB1201500), the National Natural Science Foundation of China (61602376, 61773313, 61602374, and 61702411), the National Natural Science Foundation of Shaanxi Province (2017JQ6020,

2016JQ6041), the Key Research and Development Program of Shaanxi Province (2020GY-039, 2017ZDXM-GY-098, and 2019TD-014).

### References

- [1] Statista, *Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2018*[EB/OL], 2018, August 2018, <https://www.statista.com/statistics/266136/global-market-share-held-bysmartphone-operating-systems/>.
- [2] MITRE, *Cve details: Android vulnerabilities*. [OL], 2018, June 2018, <https://www.cvedetails.com/product/19997/Google-Android.html>.
- [3] M. Linaresvasquez, G. Bavota, and C. Escobarvelasquez, "An empirical study on android-related vulnerabilities," *14th International Conference on Mining Software Repositories*, pp. 2–13, 2017.
- [4] Z. Xiaojing, "An autonomous protection algorithm for android malware attacks based on multiple features," *Proceedings of 2019 International Conference on Information Science, Medical and Health Informatics (ISMHI 2019)*. Institute of Management Science and Industrial Engineering, pp. 573–576, 2019.
- [5] G. Ye, Z. Tang, D. Fang et al., "A video-based attack for Android pattern lock," *ACM Transactions on Privacy and Security (TOPS)*, vol. 21, no. 4, 2018.
- [6] M. Youn-A, C. Tae-Mu, and J. M. Kim, "A study on Android attack by drive Management," *Advanced Science Letters*, vol. 23, no. 10, pp. 9926–9929, 2017.
- [7] B. Kong, L. Ying, and L.-P. Ma, "PtmxGuard: An Improved Method for Android Kernel to Prevent Privilege Escalation attack," *ITM Web of Conferences*, vol. 12, p. 05010, 2017.
- [8] A. H. N. Woo Hyun, P. A. R. K. Sanghyeon, O. H. Jaewon, and L. I. M. Seung-Ho, "Inishing: a UI phishing attack to exploit the vulnerability of inotify in Android smartphones," *The*

*Institute of Electronics, Information and Communication Engineers*, vol. E99, 2016.

- [9] J. Gu, C. Li, D. Lei, and Q. Li, "Combination attack of android applications analysis scheme based on privacy leak," in *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, Beijing, China, 2016.
- [10] F. M. Faqiry, R. Rahman, and D. S. Tomar, "Scrutinizing permission based attack on android os platform devices," *International Journal*, vol. 8, no. 7, 2017.
- [11] W. Bao, W. Yao, M. Zong, and D. Wang, "Cross-site scripting attacks on Android hybrid applications," *Proceedings of the 2017 International Conference on Cryptography, Security and Privacy*, pp. 56–61, 2017.
- [12] S. Heuser, M. Negro, P. K. Pendyala, and A.-R. Sadeghi, "Droid auditor: forensic analysis of application-layer privilege escalation attacks on Android," *Proceedings of the 20th International Conference on Financial Cryptography and Data Security*, 2016.
- [13] J. Vila and R. J. Rodríguez, "Practical experiences on NFC relay attacks with android," in *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, pp. 87–103, Springer International Publishing, 2015.
- [14] M. Kato and S. Matsuura, "A dynamic countermeasure method to android malware by user approval," in *Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual*, pp. 730–731, Kyoto, Japan, July 2013.
- [15] S. Y. Shin, Y. W. Kang, and Y. G. Kim, "Android-GAN: Defending against android pattern attacks using multi-modal generative network as anomaly detector. Expert Systems with Applications," *Journal of Engineering*, vol. 141, Article ID 112964, 2020.
- [16] S.-Y. Shin, Y.-W. Kang, and Y.-G. Kim, "Android-GAN: defending against android pattern attacks using multi-modal generative network as anomaly detector," *Expert Systems with Applications*, vol. 141, p. 112964, 2020.
- [17] S. Xinlong, "Mobile device management system based on AOSP and SELinux," in *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pp. 111–114, Shenzhen, China, June 2017.
- [18] M. Lange, S. Liebergeld, A. Lackorzynski, A. Warg, and M. Peter, "L4Android: a generic operating system framework for secure smartphones," *Proceedings of the 1st ACM Workshop on security and Privacy in Smartphones and Mobile Devices*, pp. 39–50, ACM, New York, 2011.
- [19] Y. Yang, Z. J. Qian, and H. Huang, "A lightweight monitor for Android kernel protection," *Computer Engineering*, vol. 40, no. 4, pp. 48–52, 2014.
- [20] L. Zicong, X. Kaiyong, G. Song, and X. Jingxu, "Dynamic measurement method of Android kernel based on ARM virtualization extension," *Computer application*, vol. 38, no. 9, pp. 2644–2649, 2018.
- [21] D. Zhang, L. Chen, F. Xue, H. Wu, and H. Huang, "T-MAC: protecting mandatory access control system integrity from malicious execution environment on ARM-based mobile devices," in *International Conference on Information Security*, pp. 348–365, Springer, Cham, 2017.
- [22] X. Ge, H. Vijayakumar, and T. Jaeger, "Sprobes: enforcing kernel code integrity on the TrustZone architecture," *Computer Science*, vol. 25, no. 6, pp. 1793–1795, 2014.
- [23] B. Lapid and A. Wool, "Cache-attacks on the ARM TrustZone implementations of AES-256 and AES-256-GCM via GPU-based analysis," *25th international conference on selected areas*, 2018.
- [24] A. M. Azab, K. Swidowski, R. Bhutkar et al., "SKEE: a lightweight secure kernel-level execution environment for ARM," in *Proceedings of Network and Distributed System Security Symposium*, San Diego, CA, USA, 2016.
- [25] R. B. Yehuda and N. J. Zaidenberg, "Protection against reverse engineering in ARM," *International Journal of Information Security*, vol. 19, no. 1, 2020.
- [26] F. Dengguo, Q. Yu, W. Dan, and C. Xiaobo, "Research on trusted computing technology," *Journal of Computer Research and Development*, vol. 48, no. 8, pp. 1332–1349, 2011.
- [27] C. X. Shen, H. G. Zhang, D. G. Feng, Z. F. Cao, and J. W. Huang, "Survey of information security," *Science in China Series F: Information Sciences*, vol. 50, no. 3, pp. 273–298, 2007.
- [28] N. Asokan, J. E. Ekberg, K. Kostiaainen et al., "Mobile trusted computing," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1189–1206, 2014.
- [29] Global Platform Device Technology, *TEE Client API Specification Version 1.0*, 2010.
- [30] C. Shuyi, W. Yingyou, and Z. Hong, "Modeling trusted computing," *Wuhan University Journal of Natural Sciences*, vol. 11, no. 6, pp. 1507–1510, 2006.

## Research Article

# An Efficient Anonymous Authentication Scheme for Mobile Pay-TV Systems

Yuting Li,<sup>1,2</sup> Qingfeng Cheng ,<sup>1,2</sup> and Jinzheng Cao<sup>1,2</sup>

<sup>1</sup>Strategic Support Force Information Engineering University, Zhengzhou 450001, China

<sup>2</sup>State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China

Correspondence should be addressed to Qingfeng Cheng; [qingfengc2008@sina.com](mailto:qingfengc2008@sina.com)

Received 22 April 2020; Revised 13 August 2020; Accepted 25 August 2020; Published 10 September 2020

Academic Editor: Ding Wang

Copyright © 2020 Yuting Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As a component of mobile communication, the pay-TV system has attracted a lot of attention. By using mobile devices, users interact with the head end system in service providers to acquire TV services. With the growth of mobile users, how to protect the privacy of users while improving efficiency of the network has become an issue worthy of attention. Anonymous authentication schemes for mobile pay-TV systems came into being. In this paper, we analyze the shortcomings of the existing authentication protocol and then propose an improved one, which is secure against stored set attack and user traceability attack. The proposed scheme is proved to be secure. Moreover, our new scheme performs better in efficiency and storage, compared with several other schemes.

## 1. Introduction

With the rapid development of wireless communication technology, pay-TV systems have attracted a lot of attention as a component of mobile communication. According to Ref. [1], the number of users who used the pay-TV system reached 3.45 million in 1994, in England. Four years later, that number has doubled. TV service is developing from socialization to personalization, which means that users are able to watch their favourite TV programs anytime, anywhere. The pay-TV systems can meet the personalized needs of users. These changes have prompted the emergence of many communication systems for mobile TV services [2, 3].

In a pay-TV system, there are two entities, a service provider and a user. When a user needs a TV service, she interacts with the head end system (HES) of the service provider. The pay-TV system generally uses a conditional access system (CAS) to handle interactions between end users and service providers. Figure 1 shows the main components of CAS, which controls the reception of TV services by encrypting transmission services to ensure that only authorized users can access certain services. The transmitter (TX)

and the receiving module (RX) are subsystems responsible for signal transmission and reception, respectively. The multiplexer (MUX) is responsible for multiplexing audio and video into the MPEG-2 transport stream, while the demultiplexer (DEMUX) is responsible for separating audio and video from the MPEG-2 transport stream. The subscriber authorization system (SAS) and subscriber management system (SMS) authorize and manage users separately.

Encryption and authentication play significant roles in CAS for mobile pay-TV systems. Obviously, we can see encryption and authentication processes Figure 1. The encryptor and the decryptor are responsible for encryption. When a user needs to obtain a service, she sends subscription and authentication messages to HES. In detail, the encryption keys must be distributed to all subscribers so that they can receive and decrypt the broadcasts they are entitled to under the terms of their subscriptions. Each receiver first filters the corresponding EMM messages and decrypts the SK and then decrypts ECM using SK. After the authorized user gets CW from ECM, she could descramble the content.

As for highly distributed mobile TV service delivery architectures [4], cloud computing models are unable to meet

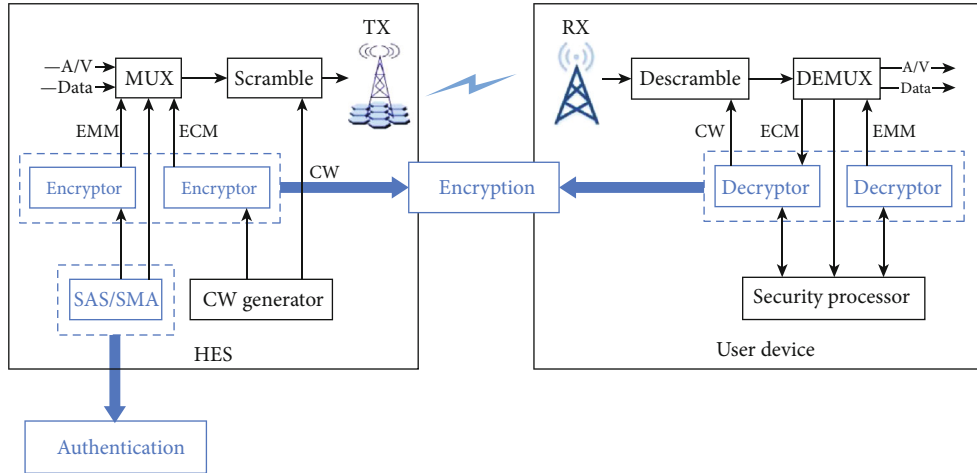


FIGURE 1: Encryption and authentication in conditional access system. Encryption and authentication process are marked in blue.

demands. The massive data generated by various access devices has made cloud network bandwidth even more limited, causing greater data bottlenecks [5]. For example, delay-sensitive business systems do not work well in cloud computing. These delay-sensitive services are often located at the edge of the data centers and can use nearby computing resources to complete calculations or reduce delays.

On the other hand, data generated by the terminal TV devices usually involves personal privacy information. Uploading these data to the cloud data center not only consumes a lot of bandwidth resources but also increases the risk of user privacy leakage [6, 7]. In order to deal with this problem, the user's identity and password are involved in anonymous authentication protocols. The role of user-generated passwords is becoming more prominent in wireless mobile networks [8]. Two-factor anonymous authentication schemes have been proposed to wireless networks for a long time [9, 10]. Moreover, three-factor authentication and key agreements have also been widely used for cloud environment [11, 12]. Besides, fuzzy commitment with low latency can also be employed to ensure high efficiency [13].

In recent years, mobile pay-TV systems have risen in popularity due to their extensive application. The most challenging issue is providing secure authentication [14]. There have been many studies on anonymous authentication schemes used for HES. In Ref. [15], Far and Alagheband designed a lightweight anonymous authentication protocol. We found that this protocol is suffering from the risk of revealing user's password. Besides, there is still room for improvement in storage. The main contributions of our paper are listed below:

- (i) We reveal Far and Alagheband's protocol is suffering from the risk of revealing user's privacy. Besides, there is still room for improvement in storage
- (ii) We propose a new efficient anonymous authentication scheme based on Far and Alagheband's protocol
- (iii) The proposed anonymous authentication scheme in the paper performs better in computing efficiency

and storage, which is more suitable for resource-constrained devices in edge computing environment

The rest of the paper is planned as follows. In Section 2, we describe related authentication schemes used in pay-TV systems. In Section 3, the preliminaries needed in protocol design are listed. The proposed anonymous authentication scheme is described in detail in Section 4. In Section 5, we give analysis of security proof and security features. Performance comparison is shown in Section 6. The conclusion is given in Section 7.

## 2. Related Work

In this section, we first introduce secure CASs and categorize pay-TV systems in three groups. Encryption-based pay-TV systems are the most classic category. Signature-based pay-TV systems are the most practical application. Authentication schemes for pay-TV systems are the most important point of our attention. Table 1 shows the relationships of some related works in chronological order.

*2.1. Secure CASs.* In 1992, ITU first proposed the standards for CASs in pay-TV systems [16]. However, this standard does not provide authentication capabilities for service providers. Since then, in order to further strengthen security, the academic community has proposed some CASs based on symmetric cryptography. In this type of CASs, users must share group keys used to encrypt and decrypt.

Zhu proposed a one-to-many CAS [17]. This system adopted the word-counting model for the first time, which improved the overall efficiency of the system to some extent. However, because the number of keys that a user needs to save was directly proportional to the number of related users, the storage and distribution of keys became very complicated, so this type of CASs was not suitable for practical applications. In general, CASs based on symmetric encryption could not avoid complicated key distribution problems. At the same time, such systems could not provide nonrepudiation.



TABLE 1: The relationship of related works.

Schemes	Year	Base article	Contribution
Song and Korba [29]	2003	Lee et al. [28]	Designed an improved version using RSA blind signature technology
Wang and Laith [21]	2008	Huang et al. [20]	Proposed an improved key distribution scheme
Sun and Leu [34]	2009	Yang and Chang [31]	Designed the first one-to-many authentication scheme
Wang and Qin [35]	2012	Sun and Leu [34]	Presented an enhanced scheme against impersonation attacks
Kim and Lee [33]	2012	Chen et al. [32]	Gave an improved version against password guessing attack and impersonation attack
Arshad et al. [36]	2017	Wang and Qin [35]	Designed an authentication scheme without bilinear pairings
Far and Alagheband [15]	2018	Chen et al. [32]	Proposed a strengthened scheme to alleviate its security risks

In 2019, Pal and Alam proposed a channel package free centralized key distribution scheme, which was based on dynamicity of the groups [18]. The scheme used finite state machine (FSM) and optimal binary search tree (OBST) data, providing leaving and joining mechanisms for both batch users and single user. Recently, Kumar et al. [19] designed a key management protocol for access control for the pay-TV system, using the theory of numbers. The protocol is said to achieve the minimum communication complexity and storage overhead.

*2.2. Encryption-Based Pay-TV Systems.* In 2004, Huang et al. divided users into different groups according to their various preferences, and each group shared the key [20]. However, Wang and Laith found that Huang et al.'s protocol was vulnerable to key leakage attack [21]. To enhance security, they proposed an improved key distribution scheme. In the same year, Sun et al. introduced a four-layer key hierarchy model, supporting more users to make flexible choices [22]. These CASs have a common feature in that one request message corresponds to one reply request, so they cannot respond to multiple requests in a short time. The one-to-many CASs, which can respond to many service requests at the same time, have become a new research direction.

In 2005, Yeung et al. constructed a new CAS based on the RSA algorithm. In their protocol, the media service provider and the proxy service provider needed to jointly encrypt the TV programs [23]. Several years later, Yeu and Huang presented an attribute-based encryption-based access control scheme and extended it with a revocation mechanism [24]. However, the scheme was pointed to be vulnerable to collusion attacks by Rial [25].

*2.3. Signature-Based Pay-TV Systems.* As one of the cryptographic primitives, signature provides the integrity and authentication of messages [26, 27]. To solve this kind of problem, Lee et al. proposed an authentication protocol based on digital signature technology [28]. However, this protocol could not provide anonymity for service providers. To strengthen its security, Song and Korba designed an improved version of the authentication protocol, using RSA blind signature technology [29]. Since then, Roh and Jung also adopted RSA-based proxy signature technology and designed a new authentication scheme [30]. However, the communication cost of their scheme was relatively high and it was not suitable for practical application.

*2.4. Authentication Schemes for Pay-TV Systems.* The authentication scheme applicable to pay-TV systems cannot be directly applied to mobile pay-TV systems. Yang and Chang designed an authentication scheme for mobile pay-TV systems using elliptic curve cryptography [31]. However, Chen et al. [32] pointed out that there were security issues in Yang and Chang's scheme and proposed an anonymous authentication protocol to solve the insecure risks. They claimed that their protocol is better for applications with low power-consuming devices and high security requirements. However, Kim and Lee showed that Chen et al.'s protocol suffers the risks in password guessing attack and impersonation attack and gave an improved version [33]. In 2018, Far and Alagheband also enhanced the security in Chen et al.'s protocol to alleviate its security risks [15].

To improve the performance, Sun and Leu designed the first one-to-many authentication scheme in 2009 [34]. The scheme also used elliptic curve cryptography, suitable for access control in mobile pay-TV systems. However, Wang and Qin found that Sun and Leu's scheme had security risks [35]. The adversary could not only pretend to be a mobile set (MS) to deceive HES but also pretend to be MS to deceive HES. Moreover, Sun and Leu's scheme could not prevent unauthorized entities from accessing mobile TV programs. In order to strengthen security, Wang and Qin proposed a strengthened authentication protocol and claimed that their protocol could resist various common attacks. Based on Wang and Qin's scheme [34], Arshad et al. designed an encryption-based authentication scheme for mobile pay-TV. This scheme did not use bilinear pairings and was easily implemented on FPGA boards [36].

In 2013, Liu and Zhang designed an identity-based encryption scheme based on bilinear pairings [37]. In addition, the batch verification technique allowed the service provider to authenticate various requests from different subscribers.

Sabzinejad et al.'s scheme was also designed using a bilinear pair in 2016 [38]. Its running time was shorter than previous solutions, but it was not suitable for lightweight devices. Kuo proposed an authentication scheme based on smart cards and biometrics for mobile pay-TV, which could be used on lightweight smart card devices for multiserver environments [39]. Wu et al. proposed an authentication scheme based on user signatures for mobile pay-TV, but this scheme could not guarantee user anonymity [40]. Zhu presented a deniable authentication protocol for pay-TV system based on chaotic maps, which is called DAP-TV [41]. In 2020,

TABLE 2: Notations of entities and parameters.

Entities	Description	Parameters	Description
HES	Head end system	$S$	The server
MS	Mobile set	$U$	The user
SAS	Subscriber authorization system	ID	Identity of the user
SMS	Subscriber management system	PW	Password of the user
CW	Control word	$b$	Random number
CAS	Conditional access system	$T$	Timestamp
ECM	Entitlement control message	$\Delta T$	Specified maximum time difference
EMM	Entitlement management message	$N$	User registration number
DBS	Data base server	$\Theta$	Token for issue phase
MUX	Multiplexer	$\gamma$	Token for subscription phase
DEMUX	Demultiplexer	$\eta$	Token for hand-off phase
TX	Transmitter	$h(\cdot)$	One-way hash function
RX	Receiving module	$x$	Secret key of DBS
DVB	Digital video broadcast	$\mathcal{A}$	The adversary

Kumaravelu et al. [14] designed an anonymous scheme which can authenticate both users and HES, with low computational cost.

### 3. System Model and Security Requirements

In this section, the operating mechanism of mobile pay-TV systems is explained at first. The security features required in anonymous authentication schemes and adversary capabilities are then briefly explained.

*3.1. Anonymous Authentication Model for Mobile Pay-TV Systems.* Table 2 shows notations of entities and parameters. The mobile pay-TV system consists of two important components, the head end system (HES) and the mobile set (MS). HES not only has powerful service content processing capabilities but also contains SAS/SMS. SAS/SMS is mainly responsible for authentication and key management, payment management, and subscription information management. MS is a user equipment that can use the mobile Internet connection to HES to obtain TV services.

In general, when a user wants to purchase a mobile pay-TV service, she needs to register the private information in HES, such as an ID number and email address. When the user needs TV services, his MS will send a request message for MS authentication and a service content request to HES. If the MS passes the HES authentication, the HES will broadcast a request message for the HES authentication to all nearby mobile sets. After the MS completes the authentication of the HES, the user can obtain service rights and enjoy the mobile pay-TV service. When the user wants to switch to another TV service, the MS and HES need to conduct mutual authentication again.

More specifically, there are four steps in the process of mobile TV and HES authentication and subscription services. In the initialization phase, DBS is responsible for generating system parameters and secret parameters required by MS. All HESs can obtain the parameters stored in DBS,

which are generated in the initialization phase. In the issue phase, MS sends a log-in request to one HES to obtain a service then authenticates with this HES. As a result, the HES will issue a token for MS, which will be used in the subscription phase to subscribe a service. When the mobile TV wants to move to another area covered by other HES, all the MS needs to do is to authenticate with the new HES, not to reregister or send a log-in request. These four steps are shown in Figure 2.

*3.2. Security Requirements.* The anonymous authentication protocols used in mobile pay-TV systems need to provide mutual authentication, forward security, and privacy protection of each entity. In addition, the importance of user anonymity and user untraceability is more emphasized in mobile pay-TV systems.

*3.2.1. Mutual Authentication.* HES and MS need to perform mutual authentication, to conduct subsequent key management, payment management, and subscription management. For resource-constrained devices, the efficiency of authentication should be taken into consideration.

*3.2.2. Forward Security.* One of the characteristics of mobile users is frequent log-in and log-out. Therefore, when a mobile user leaves a communication network, others cannot infer any user information from the encrypted message left by the user. Forward security means that the authenticated keys generated from each session are independent of each other.

*3.2.3. User Anonymity.* User anonymity is the most basic requirement in an anonymous authentication protocol, which hides the user's identity and communication relationship in the communication process through a certain method. This usually means that the user's identity cannot be obtained by anyone, whether he is an internal attacker or an external attacker. In other words, the identity of the user cannot be publicly transmitted in plaintext.

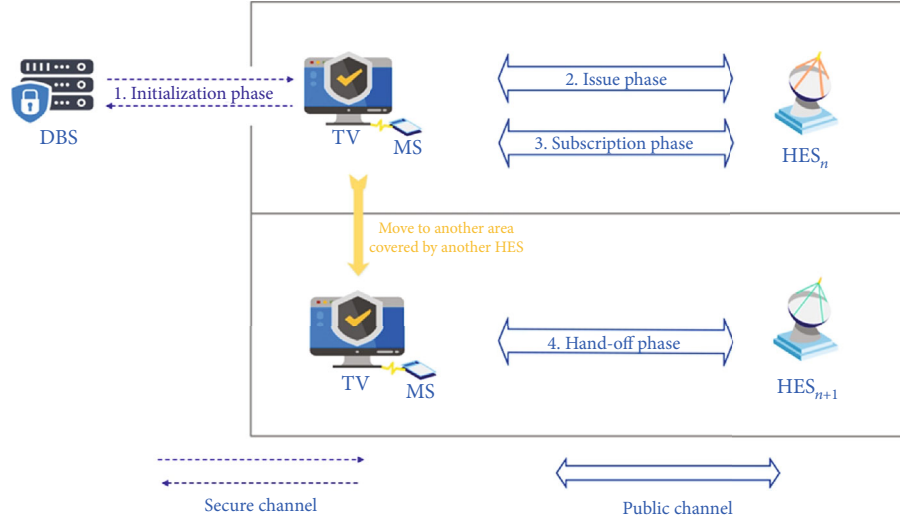


FIGURE 2: Anonymous authentication model for mobile pay-TV system. The initialization phase is performed on secure channel, while the other three phases can be performed on public channel.

**3.2.4. User Untraceability.** User untraceability has many implications. Malicious attackers or other users cannot determine which servers a user has logged in to or how many times a user has logged in to a server. Untraceability can ensure that even if the user reveals his identity at a certain stage, it will not help the adversary to identify the user at other stages. An effective way to achieve untraceability is to randomize the information transmitted in each step of the authentication phase.

**3.2.5. Privacy Protection.** Privacy protection means that the information of both MS and HES should be unavailable to others. In mobile pay-TV systems, the user logs in anonymously and does not want anyone to know her identity information. This requires that the identity information cannot be stored and transmitted in plain text.

**3.3. Adversary Capabilities.** As defined in other anonymous authentication protocols for mobile pay-TV systems, adversaries have the ability to do all passive attacks, such as eavesdropping on messages in public channel. Moreover, the adversary is allowed to obtain all parameters stored in DBS.

In order to prove that our scheme has more advantages in security, we have given adversaries the ability to obtain stored sets. That means the information stored in smart cards of MS and HES is not secure anymore.

The capabilities of adversaries are described briefly below:

- (i)  $\mathcal{A}$  can eavesdrop on messages in public channel
- (ii)  $\mathcal{A}$  can obtain all parameters stored in DBS
- (iii)  $\mathcal{A}$  can achieve all information stored in stored set of MS
- (iv)  $\mathcal{A}$  can be a internal attacker

## 4. The Proposed Scheme

In this section, we explain an improved scheme of Far and Alagheband's scheme. Our improved scheme also has four phases as depicted in Section 3, the initialization phase, issue phase, subscription phase, and hand-off phase. The initialization phase is performed on secure channel, while the other three phases can be performed on public channel. These four phases are described, respectively, as below. The notations used in this section are shown in Table 2.

**4.1. Initialization Phase.** In the initialization phase, the MS should register in SAS/SMS through DBS, which stores data in HES. This phase needs to be performed on a secure channel. More details are listed as follows.

**MS:** chooses a random number  $b$  and generates its password  $PW$ , then computes  $PWB = h(PW||b)$ . After that, it sends  $ID$  and  $PW$  to DBS of  $HES_n$ .

**DBS:** after receiving  $ID$  and  $PW$  from the MS, DBS computes  $Q = h(ID||x) \oplus PWB$ ,  $R = h(PWB||ID) \oplus h(ID||x)$ , and  $t = h(PWB||h(ID||x))$ . Here,  $x$  is the secret key of the DBS, which is generated by  $HES_n$ . Finally, DBS stores  $R$  and  $t$ , then sends  $Q$  and  $R$  to MS.

**MS:** after receiving  $Q$  and  $R$  from DBS, MS stores  $Q$  and  $R$

The initialization phase is shown in Figure 3.

**4.2. Issue Phase.** Before a mobile TV wants to obtain a service, the MS needs to send a service start request to  $HES_n$ , that is, log-in request. After sending a log-in request, MS and  $HES_n$  authenticate each other in the issue phase. As a result,  $HES_n$  will issue a token for MS, which will be used in the subscription phase. The detailed authentication process is described in Figure 4.

**MS:** computes  $PWB = h(PW||b)$  and verifies  $R = Q \oplus PWB \oplus h(PWB||ID)$ . If verified, it then computes  $W = h(h(PWB||Q) \oplus PWB) \oplus T_1$ ,  $CID = W \oplus h(W||T_1)$ ,  $C = h(R||CID||T_1)$ ,

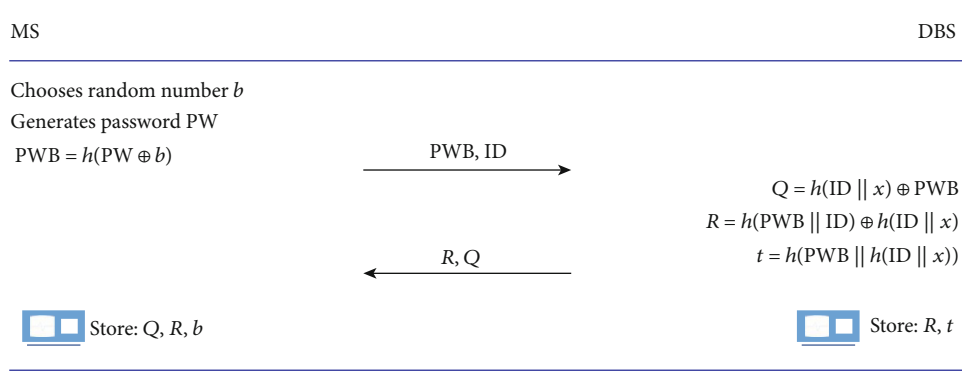


FIGURE 3: Initialization phase. The initialization phase needs to be performed on a secure channel.

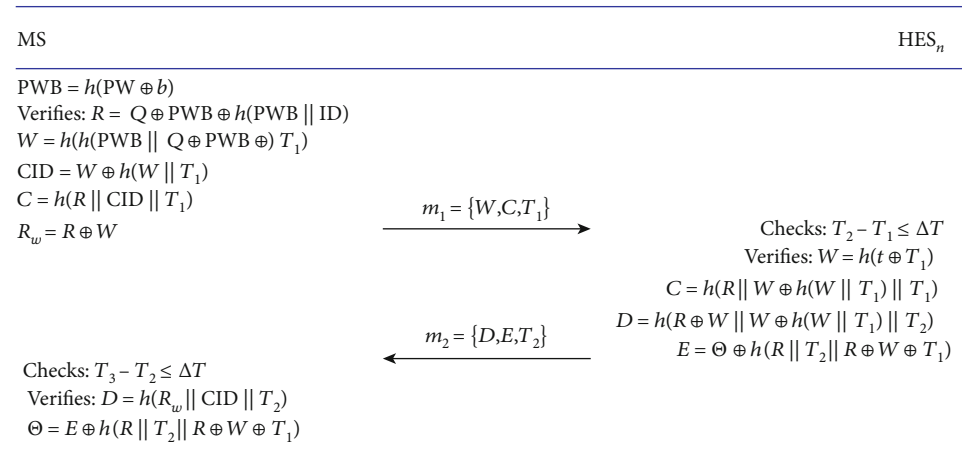


FIGURE 4: Issue phase. The issue phase can be performed on public channel.

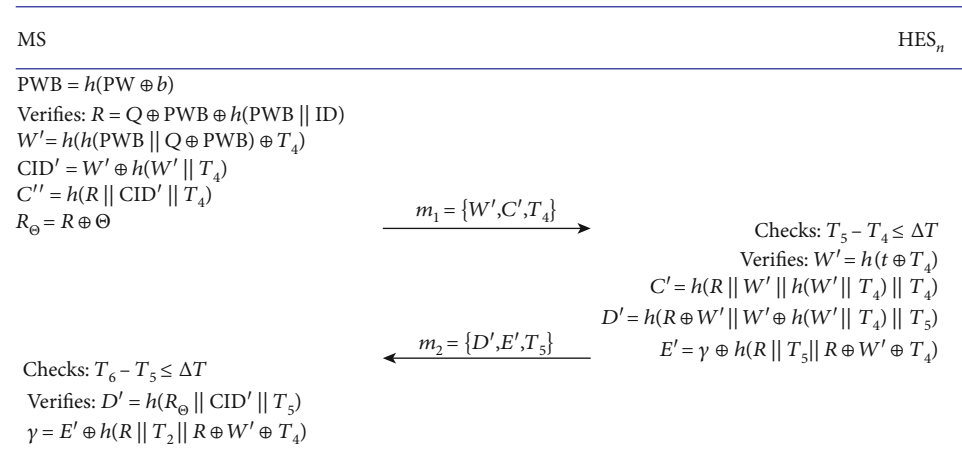


FIGURE 5: Subscription phase. The subscription phase can be performed on public channel.

and  $R_w = R \oplus W$ , and finally sends  $m_1 = \{W, C, T_1\}$  to HES<sub>n</sub> at  $T_1$ .

HES<sub>n</sub>: receives message at  $T_2$ . It first checks  $T_2 - T_1 \leq \Delta T$ , then verifies  $W = h(t \oplus T_1)$  and  $C = h(R \parallel W \oplus h(W \parallel T_1) \parallel T_1)$ . Next, it chooses a token  $\Theta$  and computes  $D = h(R \oplus W \parallel W \oplus h(W \parallel T_1) \parallel T_2)$ ,  $E = \Theta \oplus h(R \parallel T_2 \parallel R \oplus W \oplus T_1)$ , and finally sends  $m_2 = \{D, E, T_2\}$  to MS at  $T_3$ .

MS: after receiving  $m_2$ , it first checks  $T_3 - T_2 \leq \Delta T$ . Then, it verifies  $D' = h(R_w \parallel \text{CID} \parallel T_2)$ . The authentication key is computed as  $\Theta = E \oplus h(R \parallel T_2 \parallel R \oplus W \oplus T_1)$ .

4.3. *Subscription Phase.* Once the MS has obtained the token from the HES<sub>n</sub>, it can use it to subscribe to the service. Except for the token  $\Theta$  from the issue phase to participate in the

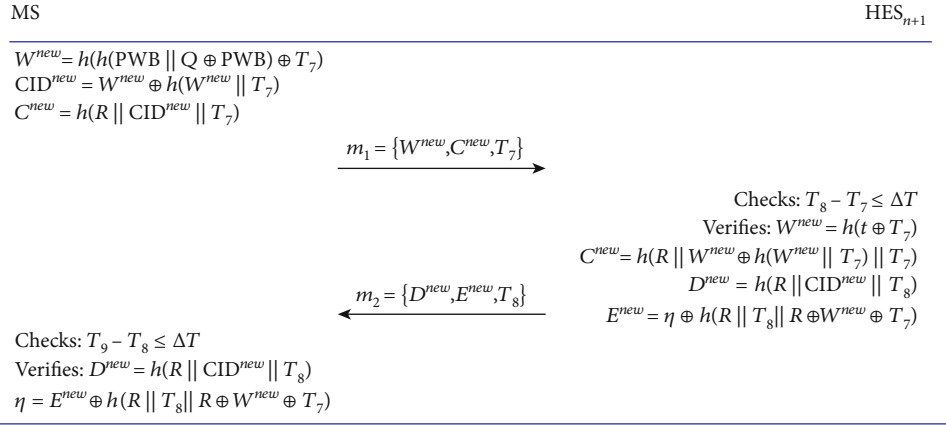


FIGURE 6: Hand-off phase. The hand-off phase can be performed on public channel.

operation, other steps are similar to the issue phase. The details are showed in Figure 5.

MS: computes  $\text{PWB} = h(\text{PW} \parallel b)$  and verifies  $R = \text{Q} \oplus \text{PWB} \oplus h(\text{PWB} \parallel \text{ID})$ . If verified, it then computes  $W' = h(h(\text{PWB} \parallel \text{Q} \oplus \text{PWB}) \oplus T_4)$ ,  $\text{CID}' = W' \oplus h(W' \parallel T_4)$ ,  $C' = h(R \parallel \text{CID}' \parallel T_4)$ , and  $R_{\ominus} = R \oplus \Theta$ , and finally sends  $m_1 = \{W', C', T_4\}$  to HES<sub>n</sub> at  $T_4$ .

HES<sub>n</sub>: receives message at  $T_5$ . It first checks  $T_5 - T_4 \leq \Delta T$ , then verifies:  $W' = h(t \oplus T_4)$ ,  $C' = h(R \parallel W' \oplus h(W' \parallel T_4) \parallel T_4)$ . Next, it chooses a new token  $\gamma$  and computes  $D' = h(R \oplus \Theta \parallel W' \oplus h(W' \parallel T_4) \parallel T_5)$  and  $E' = \gamma \oplus h(R \parallel T_5 \parallel R \oplus W' \oplus T_4)$  and finally sends  $m_2 = \{D', E', T_5\}$  to MS at  $T_6$ .

MS: after receiving  $m_2$ , it first checks  $T_6 - T_5 \leq \Delta T$ , then verifies  $D' = h(R_{\ominus} \parallel \text{CID}' \parallel T_5)$ . The authentication key is computed as  $\gamma = E' \oplus h(R \parallel T_5 \parallel R \oplus W' \oplus T_4)$ .

**4.4. Hand-Off Phase.** When a mobile user wants to move from the area covered by HES<sub>n</sub> to another area covered by HES<sub>n+1</sub>, he does not need to reregister or send a log-in request. All the MS needs to do is to authenticate with the new HES<sub>n+1</sub>. The details are showed in Figure 6.

MS: first computes  $W^{new} = h(h(\text{PWB} \parallel \text{Q} \oplus \text{PWB}) \oplus T_7)$ ,  $\text{CID}^{new} = W^{new} \oplus h(W^{new} \parallel T_7)$ , and  $C^{new} = h(R \parallel \text{CID}^{new} \parallel T_7)$ , and then sends  $m_1 = \{W^{new}, C^{new}, T_7\}$  to HES<sub>n+1</sub> at  $T_7$ .

HES<sub>n+1</sub>: receives message at  $T_8$ . It first checks  $T_8 - T_7 \leq \Delta T$ , then verifies  $W^{new} = h(t \oplus T_7)$  and  $C^{new} = h(R \parallel W^{new} \oplus h(W^{new} \parallel T_7) \parallel T_7)$ . Next, it chooses a new token  $\eta$  and computes  $D^{new} = h(R \parallel \text{CID}^{new} \parallel T_8)$ ,  $E^{new} = \eta \oplus h(R \parallel T_8 \parallel W^{new} \oplus R \oplus T_7)$ . Finally, it sends  $m_2 = \{D^{new}, E^{new}, T_8\}$  to MS at  $T_9$ .

MS: after receiving  $m_2$ , it first checks  $T_9 - T_8 \leq \Delta T$ , then verifies  $D^{new} = h(R \parallel \text{CID}^{new} \parallel T_8)$ . The authentication key to get services for new HES is set as  $\eta = E^{new} \oplus h(R \parallel T_8 \parallel R \oplus W^{new} \oplus T_7)$ .

## 5. Security Analysis

Security analysis is composed of two subsections. First, we prove our improved scheme to be secure using the formal method in Section 5.1. Then, the main security features in our scheme are shown in Section 5.2.

**5.1. Formal Security Analysis.** In this subsection, we will show that our improved scheme can resist eavesdropping attack, stored set attack, and internal attack. The approaches proposed in literature [15, 42, 43] are employed in this part. The adversary capabilities are given in Section 3.

First, we give the definition that the adversary successfully breaks the scheme [42]. The first thing is to explain notations:

- (i) Experiment function (EXP):  $\mathcal{A}$  successfully obtains the required information
- (ii) Success function (Succ):  $\mathcal{A}$ 's probability of success in obtaining the key secret information

**Definition 1.** If the probability of success is negligible, the scheme is secure against assumed  $\mathcal{A}$ .

$$\text{Succ} = \Pr [\text{EXP}^{\text{function}}] \leq \epsilon. \quad (1)$$

**Theorem 2.** The adversary  $\mathcal{A}$  eavesdrop on messages in public channel.  $\mathcal{A}$  can break the scheme with probability  $\Pr [\text{EXP}^{\text{hash}}] \leq \epsilon$ , where  $\epsilon$  is negligible.

**Proof of Theorem 1.**  $\mathcal{A}$  can eavesdrop  $m_1 = \{W, C, T_1\}$  in public channel. We describe the subsequent actions of  $\mathcal{A}$  in Algorithm 1, which consists of set up, challenge, and guess.

It is obviously to see that  $\mathcal{A}$  must correctly guess the value of ID,  $x$ , PW,  $b$  to pass the algorithm. The probability of correctly guessing these four values is less than  $(1/2)^{\text{length}}$ :

$$\text{Succ}_{\text{ID}} = \Pr [\text{EXP}^{\text{hash-ID}}] \leq \left(\frac{1}{2}\right)^{\text{ID-length}}, \quad (2)$$

$$\text{Succ}_x = \Pr [\text{EXP}^{\text{hash-x}}] \leq \left(\frac{1}{2}\right)^{x\text{-length}}, \quad (3)$$

Set up: Input  $\{W, C, T_1\}$  eavesdropped from public channel. If success, output 1. Otherwise, output 0.  
 Challenge:  
 (i) Eavesdrop  $\{W, C, T_1\}$  from public channel  
 (ii) Compute  $W = h(t \oplus T_1)$ . Here  $t = h(PWB \| h(ID \| x))$ ,  $PWB = h(PW \| b)$ .  
 (iii) Choose randomly  $ID^*, x^*, PW^*, b^*$  as the value of  $ID, x, PW, b$   
 (iv) Compute  $h(h(h(PW^* \| b^*) \| h(ID^* \| x^*)) \oplus T_1) = W^*$   
 Guess: If  $W^* = W$ , accept the value of  $ID^*, x^*, PW^*, b^*$ . Return 1. Otherwise, return 0.

ALGORITHM 1

Set up: Input  $R, b$  corrupted from MS. If success, output 1. Otherwise, output 0.  
 Challenge:  
 (i) Corrupt  $R, b$   
 (ii) Choose randomly  $PW^*, ID^*, x^*$  as the value of user's password, identity and server's secret key  
 (iii) Compute  $R^* = h(h(PW^* \| b) \| ID^*) \oplus h(ID^* \| x^*)$ .  
 Guess: If  $R^* = R$ , accepts the value of  $PW^*, ID^*, x^*$ . Return 1. Otherwise, returns 0.

ALGORITHM 2

Set up: Input  $Q, b$  corrupted from MS. If success, output 1. Otherwise, output 0.  
 Challenge:  
 (i) Corrupt  $Q, b$   
 (ii) Choose randomly  $PW^*, ID^*, x^*$  as the value of user's password, identity, and server's secret key  
 (iii) Compute  $Q^* = h(ID^* \oplus x^*) \oplus h(PW^* \| b)$   
 Guess: If  $Q^* = Q$ , accepts the value of  $PW^*, ID^*, x^*$ . Return 1. Otherwise, returns 0.

ALGORITHM 3

$$\text{Succ}_{PW} = \Pr \left[ \text{EXP}^{\text{hash}-PW} \right] \leq \left( \frac{1}{2} \right)^{PW\text{-length}}, \quad (4)$$

$$\text{Succ}_b = \Pr \left[ \text{EXP}^{\text{hash}-b} \right] \leq \left( \frac{1}{2} \right)^{b\text{-length}}. \quad (5)$$

Thus,  $\mathcal{A}$  can break the scheme with probability:  $\Pr \left[ \text{EXP}^{\text{hash}} \right] \leq (1/2)^{(ID \| x \| PW \| b)\text{-length}} \leq \varepsilon$ , where  $\varepsilon$  is negligible.

**Theorem 3.** *The adversary  $\mathcal{A}$  can achieve the stored set of MS.  $\mathcal{A}$  can break the scheme with probability  $\Pr \left[ \text{EXP}^{\text{hash}} \right] \leq \varepsilon$ , where  $\varepsilon$  is negligible.*

*Proof of Theorem 3.*  $\mathcal{A}$  can achieve the stored set of MS. We describe the subsequent actions of  $\mathcal{A}$  in Algorithm 2 and Algorithm 3, which represents the situation when  $\mathcal{A}$  obtains  $R, b$  and  $Q, b$ , respectively.

The key to successfully passing Algorithm 2 is to correctly guess the value of  $PW^*, ID^*, x^*$ . The probability of correctly guessing these four values is less than  $(1/2)^{\text{length}}$ :

$$\text{Succ}_{PW} = \Pr \left[ \text{EXP}^{\text{hash}-PW} \right] \leq \left( \frac{1}{2} \right)^{PW\text{-length}}, \quad (6)$$

$$\text{Succ}_{ID} = \Pr \left[ \text{EXP}^{\text{hash}-ID} \right] \leq \left( \frac{1}{2} \right)^{ID\text{-length}}, \quad (7)$$

$$\text{Succ}_x = \Pr \left[ \text{EXP}^{\text{hash}-x} \right] \leq \left( \frac{1}{2} \right)^{x\text{-length}}. \quad (8)$$

Thus,  $\mathcal{A}$  can break the scheme with probability:  $\Pr \left[ \text{EXP}^{\text{hash}} \right] \leq (1/2)^{(PW \| ID \| x)\text{-length}} \leq \varepsilon$ , where  $\varepsilon$  is negligible.

The key to successfully passing Algorithm 3 is to correctly guess the value of  $PW^*, ID^*, x^*$ . The probability of correctly guessing these four values is less than  $(1/2)^{\text{length}}$ :

$$\text{Succ}_{PW} = \Pr \left[ \text{EXP}^{\text{hash}-PW} \right] \leq \left( \frac{1}{2} \right)^{PW\text{-length}}, \quad (9)$$

$$\text{Succ}_{ID} = \Pr \left[ \text{EXP}^{\text{hash}-ID} \right] \leq \left( \frac{1}{2} \right)^{ID\text{-length}}, \quad (10)$$

$$\text{Succ}_x = \Pr \left[ \text{EXP}^{\text{hash}-x} \right] \leq \left( \frac{1}{2} \right)^{x\text{-length}}. \quad (11)$$

Thus,  $\mathcal{A}$  can break the scheme with probability:  $\Pr \left[ \text{EXP}^{\text{hash}} \right] \leq (1/2)^{(PW \| ID \| x)\text{-length}} \leq \varepsilon$ , where  $\varepsilon$  is negligible.

**Theorem 4.** *The adversary  $\mathcal{A}$  be an internal attacker.  $\mathcal{A}$  can break the scheme with probability  $\Pr \left[ \text{EXP}^{\text{hash}} \right] \leq \varepsilon$ , where  $\varepsilon$  is negligible.*

Set up: Input  $\{W, C, T_1\}$  eavesdropped from public channel. If success, output 1. Otherwise, output 0.  
 Challenge:  
 (i) Receive  $\{W, C, T_1\}$  from public channel  
 (ii) Searches  $R$  and  $t$ , where  $t = h(\text{PWB} \| h(\text{ID} \| x))$   
 (iii) Choose randomly  $\text{PW}^*, \text{ID}^*, b^*, x^*$   
 (iv) Compute  $R^* = h(h(\text{PW}^* \| b^*) \| \text{ID}^* \| h(\text{ID}^* \| x^*))$ ,  $t^* = h(h(\text{PW}^* \| b^*) \| h(\text{ID}^* \| x^*))$   
 Guess: If  $R^* = R$  or  $t^* = t$ , accepts the value of  $\text{ID}^*, b^*$ . Return 1. Otherwise, returns 0.

ALGORITHM 4

*Proof of Theorem 4.*  $\mathcal{A}$  can be a malicious server, as an internal attacker. Even so,  $\mathcal{A}$  has no way of knowing identity of the user. We describe the subsequent behavior of  $\mathcal{A}$  in Algorithm 4.

Since the hash functions we use are one-way secure, if  $\mathcal{A}$  wants to know the value of  $\text{ID}$ ,  $b$  to pass the algorithm, they can only guess. The probability of correctly guessing these two values is less than  $(1/2)^{\text{length}}$ :

$$\text{Succ}_{\text{PW}} = \Pr \left[ \text{EXP}^{\text{hash-PW}} \right] \leq \left( \frac{1}{2} \right)^{\text{PW-length}}, \quad (12)$$

$$\text{Succ}_{\text{ID}} = \Pr \left[ \text{EXP}^{\text{hash-ID}} \right] \leq \left( \frac{1}{2} \right)^{\text{ID-length}}, \quad (13)$$

$$\text{Succ}_b = \Pr \left[ \text{EXP}^{\text{hash-b}} \right] \leq \left( \frac{1}{2} \right)^{b\text{-length}}, \quad (14)$$

$$\text{Succ}_x = \Pr \left[ \text{EXP}^{\text{hash-x}} \right] \leq \left( \frac{1}{2} \right)^{x\text{-length}}. \quad (15)$$

Therefore,  $\mathcal{A}$  can break the scheme with probability:  $\Pr [\text{EXP}^{\text{hash}}] \leq (1/2)^{(\text{ID} \| b)\text{-length}} \leq \epsilon$ , where  $\epsilon$  is negligible.

In summary, our improved scheme can resist eavesdropping attack, stored set attack, and internal attack.

**5.2. Security Features.** In this subsection, we first explain the main changes in our improved scheme compared with Far and Alagheband's scheme.

(i) Bind  $x$  to  $R$  and  $Q$

In the initialization phase of Far and Alagheband's protocol,  $R$  and  $Q$  are stored directly in DBS. The user's identity is hidden in  $R$  and  $Q$  so that the user does not need to reveal its identity when logging in and out. However, there are security risks in storing  $R$ ,  $Q$ , and  $Q \oplus \text{PWB}$  in the DBS. As long as the adversary reveals DBS, she can obtain PWB by exclusive OR. This not only brings the leakage of user identity but also causes the risk of user untraceability. In our new scheme, we add the server's secret key  $x$  and make slight changes when calculating  $R$  and  $Q$ . Thus, the adversary can no longer recover user's privacy information through data in DBS.

(ii) Remove the random numbers  $n$  in the issue phase and subscription phase

TABLE 3: Notations of entities and parameters.

Security features	Far and Alagheband's scheme	The improved scheme
Mutual authentication	Yes	Yes
Forward secrecy	Yes	Yes
User anonymity	Yes	Yes
User untraceability	No	Yes
Privacy protection	No	Yes
Stored set attack	No	Yes

TABLE 4: Comparison of operation numbers in each scheme.

Schemes	A	B	C	D	E	F	G	H
The scheme in [32]	6	0.78	7	0.91	7	0.91	4	5
The scheme in [33]	7	0.91	20	1.3	7	0.91	4	5
The scheme in [15]	3	0.39	6	0.78	4	0.52	3	4
Our scheme	3	0.39	6	0.78	4	0.52	3	3

Note:  $A$ : the number of hash operations in the initialization phase.  $B$ : the execution time of the initialization phase ( $\mu\text{s}$ ).  $C$ : the number of hash operations by user side in the issue phase.  $D$ : the execution time by user side of the issue phase ( $\mu\text{s}$ ).  $E$ : the number of hash operations by server side in the issue phase.  $F$ : the execution time by server side of the issue phase ( $\mu\text{s}$ ).  $G$ : the number of parameters in the stored set.  $H$ : the number of parameters transmitted on public channel.

The introduction of random numbers is to ensure that the authentication keys generated by each session are independent of each other, in order to meet the forward security of the anonymous authentication protocol. In Far and Alagheband's protocol, the random numbers  $n$  is used. Actually, each time a session generates an authentication key, a time stamp is required. Here, the time stamp  $T_i (i = 1, \dots, 6)$  not only provides the function of mutual authentication, but also introduces freshness. Therefore, our scheme can still guarantee forward security without using random numbers.

As a result of the changes, the security of the new scheme has been improved in terms of user untraceability and privacy protection. Table 3 shows the comparison of our improved scheme and Far and Alagheband's scheme.

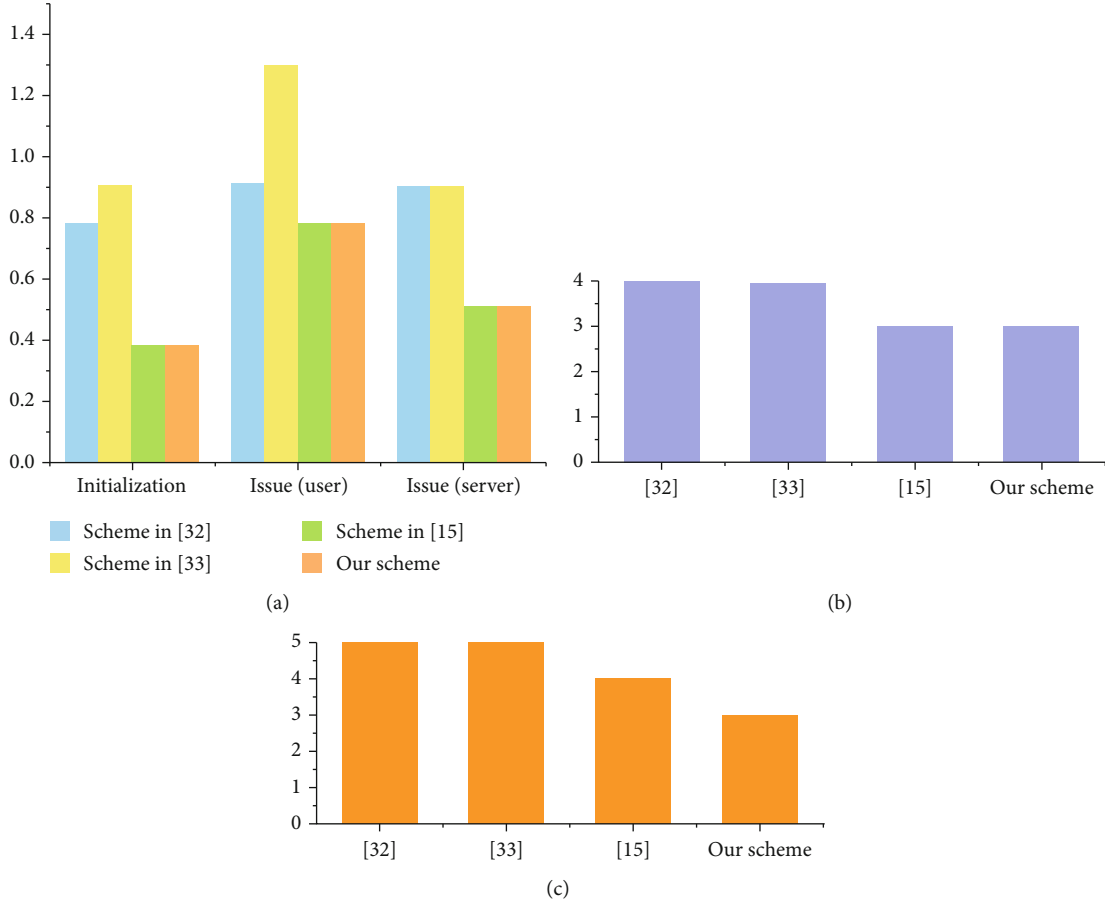


FIGURE 7: Comparison of execution time and parameter numbers. (a) Comparison of execution time. (b) Comparison of parameters in the stored set. (c) Comparison of parameters transmitted on public channel.

**5.2.1. Mutual Authentication.** In each session, HES and MS must first perform mutual authentication, using the pre-assigned  $R$ ,  $Q$ , and  $t$ . We bind the server's secret key  $x$  and the user's identity  $ID$  when calculating  $R$ ,  $Q$ , and  $t$ , to ensure the confidentiality of them. The one-way hash function also provides an efficient method for mutual authentication.

**5.2.2. Forward Security.** Forward security means that the authenticated keys generated from each session are independent of each other. In our new scheme, the time stamps  $T_i$  ( $i = 1, \dots, 6$ ) introduce the freshness of each session. Different  $T_i$  ( $i = 1, \dots, 6$ ) participating in the operation will generate different authentication keys.

**5.2.3. User Anonymity.** User anonymity means that the user's identity  $ID$  cannot be obtained by internal attackers or external attackers. In our new scheme, the identity  $ID$  of the user is not be publicly transmitted in plaintext, while it is placed in a hash function. Moreover, the server has no access to recover the user's identity  $ID$  from  $R$  and  $t$  stored in DBS.

**5.2.4. User Untraceability.** In our scheme, all HESs can obtain  $R$  and  $t$  stored in DBS when they need them. Thus, the adversary can no longer determine whether the user has logged in, by comparing the stored set of each HES. Moreover,

messages  $m_1, m_2$  transmitted in public channel are diverse from each other.

**5.2.5. Privacy Protection.** In our new scheme, we add the server's secret key  $x$  and make slight changes when calculating  $R$  and  $Q$ . Thus, the adversary can no longer recover user's privacy information through data in DBS. The proposed scheme can provide user privacy protection.

## 6. Performance Comparison

Various anonymous authentication schemes have been presented in recent years. In this section, we choose a few schemes that use only hash functions and compare them with our scheme in terms of execution efficiency.

We define the execution time of one hash operation is  $0.13 \mu s$  according to Ref. [36]. The number of hash operations of each scheme is shown in Table 4. Since the subscription phase and hand-off phase are similar with the issue phase, we only compare hash operations in the initialization phase and issue phase.

From Table 4, our scheme performs better in terms of execution time. Moreover, the number of parameters transmitted on public channel is minimal, which means our scheme performs better in computing storage. In order to



show the comparison of execution efficiency more clearly, we show the execution time in  $\mu\text{s}$  and parameter numbers in Figure 7. It is obvious to see that our scheme has the shortest execution time under the same conditions.

## 7. Conclusion

The security of pay-TV systems is facing the challenge of explosive growth of users and service content. To prevent unauthorized access in mobile pay-TV systems, anonymous authentication technologies are commonly used for secure media delivery and channel protection. In this paper, we review Far and Alagheband's protocol and find that this protocol is suffering from risks of revealing user's privacy. Besides, there is still room for improvement in storage. We alleviate the security risks of Far and Alagheband's protocol. Our improved scheme can resist stored set attack and user traceability attack. Performance comparison shows that our scheme performs better in terms of execution time and storage, which means it is suitable for resource-constrained devices in edge computing environment.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant 61872449).

## References

- [1] N. L. Rayan and K. Chaitanya, *A survey on mobile wireless networks*, International Journal of Scientific and Engineering Research, 2014.
- [2] M. Armstrong, "Competition in the pay-TV market," *Journal of the Japanese and International Economies*, vol. 13, no. 4, pp. 257–280, 1999.
- [3] A. Zakerolhosseini and M. Nikooghadam, "Secure transmission of mobile agent in dynamic distributed environment," *Wireless Personal Communications*, vol. 70, no. 2, pp. 641–656, 2013.
- [4] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog Computing for the Internet of Things: Security and Privacy Issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.
- [5] S. Yangui, P. Ravindran, O. Bibani, R. H. Glitho, and P. A. Polakos, *A platform as-a-service for hybrid cloud/fog environments*, IEEE International Symposium on Local and Metropolitan Area Networks, Rome, Italy, 2016.
- [6] J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-Preserved Pseudonym Scheme for Fog Computing Supported Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2627–2637, 2018.
- [7] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2018.
- [8] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's Law in Passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.
- [9] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.
- [10] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding Node Capture Attacks in User Authentication Schemes for Wireless Sensor Networks," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.
- [11] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K. R. Choo, "Unified Biometric Privacy Preserving Three-factor Authentication and Key Agreement for Cloud-assisted Autonomous Vehicles," *IEEE Transactions on Vehicular Technology*, p. 1, 2020.
- [12] Q. Jiang, M. K. Khan, X. Lu, J. Ma, and D. He, "A privacy preserving three-factor authentication protocol for e-Health clouds," *The Journal of Supercomputing*, vol. 72, no. 10, pp. 3826–3849, 2016.
- [13] Q. Jiang, Z. Chen, J. Ma, X. Ma, J. Shen, and D. Wu, "Optimized Fuzzy Commitment based Key Agreement Protocol for Wireless Body Area Network," *IEEE Transactions on Emerging Topics in Computing*, 2019.
- [14] R. Kumaravelu, R. Sadaiyandi, A. Selvaraj, J. Selvaraj, and G. Karthick, "Computationally efficient and secure anonymous authentication scheme for IoT-based mobile pay-TV systems," *Computational Intelligence*, vol. 36, no. 3, pp. 994–1009, 2020.
- [15] S. B. Far and M. R. Alagheband, "Analysis and improvement of a lightweight anonymous authentication protocol for mobile pay-TV systems," in *9th International Symposium on Telecommunications (IST)*, pp. 466–473, 2018.
- [16] *Conditional-access broadcasting system*, ITU-R Rec, 1992, <https://www.itu.int/rec/R-REC-BT/en>.
- [17] W. T. Zhu, "A cost-efficient secure multimedia proxy system," *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 1214–1220, 2008.
- [18] O. Pal and B. Alam, "Efficient and secure conditional access system for pay-TV systems," *Multimedia Tools and Applications*, vol. 78, no. 13, pp. 18835–18853, 2019.
- [19] V. Kumar, R. Kumar, and S. K. Pandey, "An effective and secure key management protocol for access control in pay-TV broadcasting systems using theory of numbers," in *In proceedings: International Conference on Computing Applications in Electrical & Electronics Engineering (ICCAEEE 2019)*, pp. 369–379, 2020.
- [20] Y.-L. Huang, S. Shieh, F.-S. Ho, and J.-C. Wang, "Efficient Key Distribution Schemes for Secure Media Delivery in Pay-TV Systems," *IEEE Transactions on Multimedia*, vol. 6, no. 5, pp. 760–769, 2004.
- [21] S.-Y. Wang and C.-S. Laih, "Efficient key distribution for access control in pay-TV systems," *IEEE Transactions on Multimedia*, vol. 10, no. 3, pp. 480–492, 2008.
- [22] H.-M. Sun, C.-M. Chen, and C.-Z. Shieh, "Flexible-pay-per-channel: a new model for content access control in pay-TV broadcasting systems," *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 1109–1120, 2008.

- [23] S. F. Yeung, J. C. S. Lui, and D. K. Y. Yau, "A multikey secure multimedia proxy using asymmetric reversible parametric sequences: theory, design, and implementation," *IEEE Transactions on Multimedia*, vol. 7, no. 2, pp. 330–338, 2005.
- [24] L. Yeu and J. Huang, "A conditional access system with efficient key distribution and revocation for mobile pay-TV systems," *Acm Transactions on Multimedia Computing*, vol. 9, no. 3, pp. 18:1–18:20, 2013.
- [25] A. Rial, "A conditional access system with revocation for mobile pay-TV systems revisited," *Information Processing Letters*, vol. 147, pp. 6–9, 2019.
- [26] H. Xiong, Y. Bao, X. Nie, and Y. I. Asoor, "Server-aided attribute-based signature supporting expressive access structures for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1013–1023, 2020.
- [27] Q. Mei, H. Xiong, J. Chen, M. Yang, S. Kumari, and M. K. Khan, "Efficient certificateless aggregate signature with conditional Privacy Preservation in IoV," *IEEE Systems Journal*, pp. 1–12, 2020.
- [28] N. Lee, C. Chang, C. Lin, and T. Hwang, "Privacy and non-repudiation on pay-TV systems," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 20–27, 2000.
- [29] R. Song and L. Korba, "Pay-TV system with strong privacy and non-repudiation protection," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 408–413, 2003.
- [30] H. Roh and S. Jung, "An authentication scheme for consumer electronic devices accessing mobile IPTV service from home networks," in *In proceedings: IEEE International Conference on Consumer Electronics (ICCE 2011)*, pp. 717–718, 2012.
- [31] J.-H. Yang and C.-C. Chang, "An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Computers and Security*, vol. 28, no. 3-4, pp. 138–143, 2009.
- [32] T.-H. Chen, Y.-C. Chen, W.-K. Shih, and H.-W. Wei, "An efficient anonymous authentication protocol for mobile pay-TV," *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1131–1137, 2011.
- [33] H. Kim and S. W. Lee, "Anonymous Authentication Protocol for Mobile Pay-TV System," in *Communications in Computer and Information Science*, vol. 339, p. 471, Springer, Berlin, Heidelberg, 2012.
- [34] H.-M. Sun and M.-C. Leu, "An efficient authentication scheme for access control in Mobile pay-TV systems," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 947–959, 2009.
- [35] H. Wang and B. Qin, "Improved one-to-many authentication scheme for access control in pay-TV systems," *IET Information Security*, vol. 6, no. 4, pp. 281–290, 2012.
- [36] H. Arshad, M. Nikooghadam, S. Avezverdi, and M. Nazari, "Design and FPGA implementation of an efficient security mechanism for mobile pay-TV systems," *International Journal of Communication Systems*, vol. 30, no. 15, 2017.
- [37] X. Liu and Y. Zhang, "A privacy-preserving acceleration authentication protocol for mobile pay-TV systems," *Security and Communication Networks*, vol. 6, no. 3, 372 pages, 2013.
- [38] M. S. Farash and M. A. Attari, "A provably secure and efficient authentication scheme for access control in mobile pay-TV systems," *Multimedia Tools and Applications*, vol. 75, no. 1, pp. 405–424, 2016.
- [39] W.-C. Kuo, H.-J. Wei, and Y.-H. Chen, "An enhanced secure anonymous authentication scheme based on smart cards and biometrics for multi-server environments," in *2015 10th Asia Joint Conference on Information Security*, Kaohsiung, Taiwan, 2015.
- [40] H.-L. Wu, C.-C. Chang, and C.-Y. Sun, "A secure authentication scheme with provable correctness for pay-TV systems," *Security and Communication Networks*, vol. 9, no. 11, 1588 pages, 2016.
- [41] H. Zhu, "A simplified deniable authentication scheme in cloud-based pay-TV system with privacy protection," *International Journal of Communication Systems*, vol. 32, no. 11, p. e3967, 2019.
- [42] Y. Lindell, "Anonymous Authentication," *Journal of Privacy and Confidentiality*, vol. 2, no. 2, pp. 35–63, 2011.
- [43] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, *Investigations of power analysis attacks on smartcards*, Smartcard 1999, Illinois, USA, 1999.

## Research Article

# Authenticator Rebinding Attack of the UAF Protocol on Mobile Devices

Hui Li <sup>1</sup>, Xuesong Pan <sup>1</sup>, Xinluo Wang,<sup>1</sup> Haonan Feng,<sup>1</sup> and Chengjie Shi<sup>2</sup>

<sup>1</sup>School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China

<sup>2</sup>Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100195, China

Correspondence should be addressed to Hui Li; [lihuill@bupt.edu.cn](mailto:lihuill@bupt.edu.cn)

Received 24 April 2020; Revised 21 June 2020; Accepted 9 August 2020; Published 1 September 2020

Academic Editor: Ding Wang

Copyright © 2020 Hui Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a novel attack named “Authenticator Rebinding Attack,” which aims at the Fast IDentity Online (FIDO) Universal Authentication Framework (UAF) protocol implemented on mobile devices. The presented Authenticator Rebinding Attack rebinds the victim’s identity to the attacker’s authenticator rather than the victim’s authenticator being verified by the service in the UAF protocol, allowing the attacker to bypass the UAF protocol local authentication mechanism by imitating the victim to perform sensitive operations such as transfer and payment. The lack of effective authentication between entities in the implementations of the UAF protocol used in the actual system causes the vulnerability to the Authenticator Rebinding Attack. In this paper, we implement this attack on the Android platform and evaluate its implementability, where results show that the proposed attack is implementable in the actual system and Android applications using the UAF protocol are prone to such attack. We also discuss the possible countermeasures against the threats posed by Authenticator Rebinding Attack for different stakeholders implementing UAF on the Android platform.

## 1. Introduction

FIDO UAF is an authentication mechanism based on public key cryptography designed for replacing password-based authentication [1], which has been criticized for its inconvenience and insecurity because it requires users and verifiers to maintain a growing list of login credentials as well as passwords. With FIDO UAF, users can first register their devices installed with a FIDO UAF stack to the online service by selecting a local authentication mechanism such as fingerprint and face recognition; then, users only need to repeat the local authentication operation instead of entering their passwords whenever they need to be authenticated by the service. Because of its convenience and security, UAF has attracted lots of attention in both the academic and industrial societies since its release. By April 2020, there have already been 436 certified FIDO UAF products in the market [2].

Recently, some researchers focus on analyzing the security of UAF and point out that FIDO UAF may face various potential security threats in the design and implementation of the protocol. Hu and Zhang formalize the UAF protocol

and propose hypothetical attacks such as misbinding attack, parallel session attack, and multiuser attack [3], but they neither elaborate on the assumptions required to perform these attacks nor give the concrete implementation of these attacks. Xenakis et al. present an informal security analysis of the UAF protocol and identify a list of vulnerabilities that can cause attacks such as intercepting switching data, imitating the user’s online service, and presenting false information to the user screen during the transaction [4]. However, they fail to provide any specific verification process for these attacks and ignore the actual factors when implementing the FIDO protocol, so some of the proposed attacks lack feasibility.

Most of the abovementioned FIDO UAF attacks are caused by the fact that the running environment of the UAF protocol can meet neither the UAF security assumptions described in the FIDO Security Reference [5] nor the requirements of the security standards provide by FIDO Certification [6] for FIDO products. Moreover, although FIDO UAF is widely used on mobile devices [2, 7], due to the openness and diversity of mobile devices, currently there is no

specific unified standard for the implementation of the UAF protocol on them, and certain FIDO UAF products cannot meet the UAF security assumptions, and their security levels are not suitable for actual scenarios. Our previous work [8] presents an attack for the implementation of the UAF protocol caused by the lack of a trusted display module on the mobile device, so the attacker may successfully tamper such displayed information as transaction data.

In consideration of the fact that Android is one of the most popular mobile operating systems and there are many certified providers of certified products on the Android platform [9, 10], we focus on analyzing the security of the UAF protocol implementation on mobile devices and propose a novel attack named “Authenticator Rebinding Attack”. The proposed Authenticator Rebinding Attack rebinds the victim’s identity to the attacker’s authenticator and allows the attacker to impersonate the victim to perform sensitive operations such as transfer and payment.

To the best of our knowledge, our work is the first to study the threat of active Authenticator Rebinding Attack of the UAF protocol on the Android platform. On the one hand, we study the actual implementation of this attack according to the different modes in the UAF protocol on mobile devices. On the other hand, we point out that the reason for this attack is the lack of effective authentication between entities in the implementations of the UAF protocol used in the real world. We also evaluate the impact of this attack by analyzing 42 FIDO UAF applications and find that 19% of the applications that call third-party UAF Client Applications are unable to resist the attack, while the other 81% applications that implement the UAF protocol inside themselves might also suffer from this attack if they run in a compromised environment.

The contributions of this paper can be summarized as follows:

- (i) We present a novel attack called Authenticator Rebinding Attack, which impersonates the victim to perform sensitive operations by rebinding the victim’s identity to the attacker’s authenticator
- (ii) We demonstrate the technical feasibility of Authenticator Rebinding Attack by giving the details of the attack on the Hebao Pay and Jingdong Finance applications
- (iii) We prove the practical significance of this attack by analyzing their security on the UAF applications mined from applications in the real world
- (iv) We present the main causes of this threat and the countermeasures against this attack for different stakeholders on implementing the UAF protocol on the Android platform

The rest of this paper is organized as follows. In Section 2, we present the architecture, trust model, and operations of the UAF protocol. In Section 3, we analyze two UAF implementation modes, i.e., Out-App Authenticator Mode and In-App Authenticator Mode. In Section 4, we present the

Authenticator Rebinding Attack under both the Out-App and In-App Authenticator Modes as well as verify such an attack on typical applications. In Section 5, we analyze the security of the actual applications using the UAF protocol to evaluate the implementability of the attack and present the main causes of such threat, as well as the countermeasures against the threat. In Section 6, we finally give our conclusions.

## 2. UAF Protocol

In this section, we introduce the architecture, trust model of the client side, and simplified operations on the Android platform of the UAF protocol.

*2.1. Architecture.* Figure 1 shows the architecture of the UAF protocol, which includes six entities—User Agent, UAF Client, UAF ASM, UAF Authenticator, Web Server, and UAF Server [11]. These entities are deployed on the User Device and the Relying Party. The User Device works as a client and interacts with the user, generates and stores the unique *Authentication Keys*, and computes and returns a response for the *challenge* from the server side. The Relying Party works as a server and initiates the challenge-response mechanism and verifies and stores the user credentials, e.g., unique *Authentication Public Keys*. The User Device and the Relying Party communicate with each other using a secure transport protocol (such as TLS/HTTPS [12]) established between the FIDO UAF Client and the Relying Party. Moreover, the internal communication between entities in the UAF protocol differs and depends on the protocol implementations [13].

The UAF Authenticator is the entity that can be inserted (such as a USB hardware device with PIN code protection) or embedded (such as a fingerprint sensor in a smartphone) into the User Device. On the Android platform, it is recommended to implement the UAF Authenticator as a module based on the TEE. The UAF Authenticator contains two kinds of asymmetric keys, a pair of *Attestation Keys* and several pairs of *Authentication Keys*. *Attestation Keys* are pre-stored in the UAF Authenticator and used in the registration operation. *Authentication Keys* are generated by the UAF Authenticator in the registration operation and used in the authentication operation.

The UAF ASM is a software interface between the UAF Client and the UAF Authenticator, which provides uniform API to the upper layer so that a UAF Client can support diverse UAF Authenticators with different biometric factors.

The UAF Client acts as the client of the UAF protocol. It interacts with diverse UAF Authenticators through the UAF ASM and UAF Server through a Relying Party. The User Agent interacts with the user and initiates the whole operation when the user enables biometric authentication.

On the Android platform, the UAF Client and the UAF ASM can be independent applications separated from the User Agent or built-in modules of the User Agent, which will be introduced in detail in Section 3. The Web Server provides the user application service and interacts with the UAF Server to transfer UAF protocol messages. The UAF Server

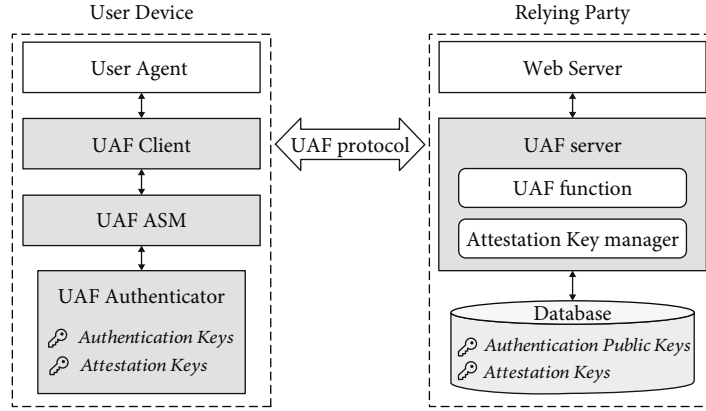


FIGURE 1: Architecture of the UAF protocol.

is responsible for communicating with the client, verifying the response message, and updating the public key related to the user. In the following section, we will use one server entity to represent the Web Server and the UAF Server to make the description more concise.

**2.2. FIDO UAF Client Trust Model.** We first introduce the FIDO UAF Client Trust Model described in FIDO UAF specification to show how these entities of the client side authenticate each other; then, we present why these authentication measures might not be effective when they are implemented on Android platform in Section 5.2.

The FIDO UAF Client Trust Model is shown in Figure 2 [14]. The FIDO UAF specification describes the data structures for authentication and access control between entities, in which *FacetID* is used for the UAF Client to authenticate the User Agent; *CallerID* is used for the UAF ASM to authenticate the UAF Client; *KHAccessToken* is used to provide access control for an *Authentication Key*. The UAF Authenticator ensures that a UAF ASM provides a specific *KHAccessToken* to access the correct user *Authentication Key*. The *KHAccessToken* is exported by the UAF ASM during the registration operation using data such as *AppID*, *PersonalID*, *ASMTOKEN*, and *CallerID* [15]. If the *AppID* received by a UAF Client is a valid HTTPS URL, the UAF Client will obtain a trusted *FacetID* list by accessing the URL (HTTPS guarantees the list is trusted), check if the *FacetID* of the User Agent is in this list and then verify the validity of the User Agent. If the *AppID* is empty, the UAF Client directly sets the *FacetID* of the User Agent to the *AppID* field and the *FacetID* will be finally verified by the server [16]. Besides, the *AAID* (Authenticator Attestation ID) identifies a model, class, or batch of UAF Authenticators that share the same characteristics. The *AAID* also identifies a pair of *Attestation (Public/Private) Keys* [17].

According to our research, the ASM-Authenticator Applications of the same version and vendor have the same *AAID* and *Attestation Keys* on the Android platform. The *FacetID* is a URI derived from the Base64 encoding SHA-1 hash of the APK signing certificate of the User Agent by the UAF Client [16]. The *CallerID* of a UAF Client is derived by the UAF ASM in the same way [15].

**2.3. UAF Protocol Operations.** The UAF protocol has two critical operations, namely, registration and authentication [13]. As shown in Figure 3, in order to describe the FIDO UAF protocol more concisely, we depict the UAF protocol operations as a challenge-response process merged from the registration and authentication operations by omitting some details.

The server and the UAF Authenticator first successfully share necessary data such as the *Attestation Public Key*, *AAID*, and protocol policies through the process of FIDO Metadata Service before the registration operation. Then, the UAF Authenticator stores its *Attestation Private Key* securely; the server sends a *challenge* to the UAF Authenticator and checks the received response while the UAF Authenticator generates a response according to the *challenge* after verifying the user's biological factors in either the registration operation or the authentication operation. The difference between these two operations is that the UAF Authenticator generates the response with the *Attestation Private Key* in the registration operation and with an *Authentication Private Key* in the authentication operation. Both the *Public Key* and the *Private Key* (in Figure 3) are referred to the *Attestation Keys* in the registration operation, as well as the *Authentication Keys* in the authentication operation. Figure 3 also shows a case where the *AppID* from the server is empty as Section 2.2 describes.

In the registration operation, the UAF Authenticator generates a pair of *Authentication Keys* associated with user profile and sends the public key signed with *Attestation Key (Private Key)* in the response message to the remote server; the server then stores the user's public key after verifying its signature by the *Attestation Public Key*; in the authentication operation, the authenticator unlocks the related *Authentication Keys* after receiving the *challenge* from the server and generates a response including a signature with *Authentication Keys (Private Key)* and sends the response message to the remote server; then, the server locates the user's public key stored in registration operation, uses it to verify the signature in the message, and finally achieves the purpose of authenticating the user's presence.

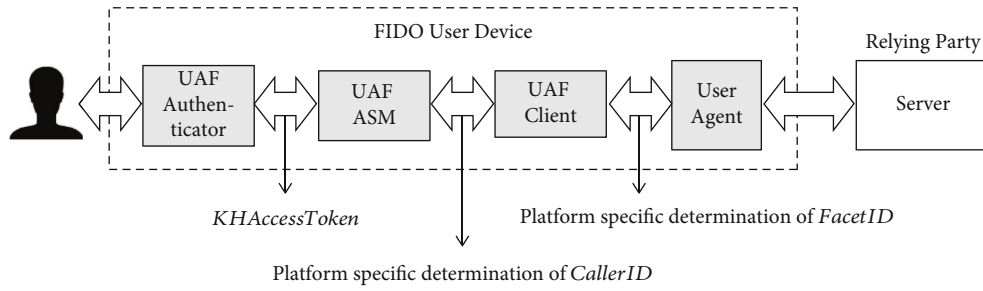


FIGURE 2: Trust Model of FIDO UAF Client.

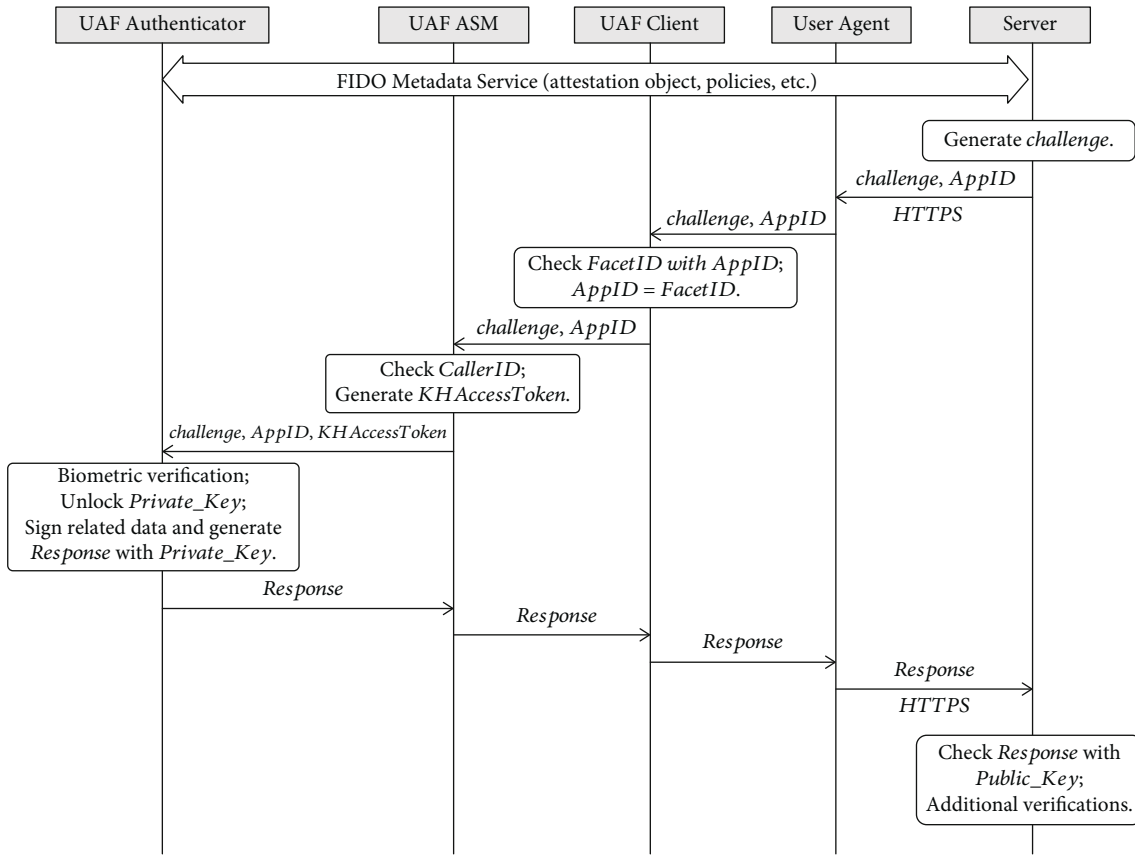


FIGURE 3: Simplified UAF protocol operation.

### 3. Implementations of the UAF Protocol

In this section, we describe two commonly implemented UAF protocol modes on the Android platform: UAF implementation based on Out-App Authenticator Mode and UAF implementation based on In-App Authenticator Mode.

**3.1. Out-App Authenticator Mode.** Out-App Authenticator Mode refers to the implementation mode where the User Agent, the UAF Client, and the ASM-Authenticator are three separate Android applications. One example is Hebao Pay, a third-party mobile payment product launched by China Mobile. [18] In the following section, we describe its implementation.

UAF Client Applications can be preinstalled in the phone by the manufacturer or installed by the user, which provide UAF Client functions that are compliant with the FIDO specifications and expose the standard interface. Upper-layer applications can implicitly call the UAF Client functions, which means that the upper-layer application and the UAF Client Application are decoupled. Therefore, an application can call different UAF Client Applications on devices of different brands without modifying their source codes. There are multiple implementations of UAF ASM and authenticators; some applications provide a UAF ASM interface to the UAF Client Application and implement the function of an authenticator at the same time through the native methods

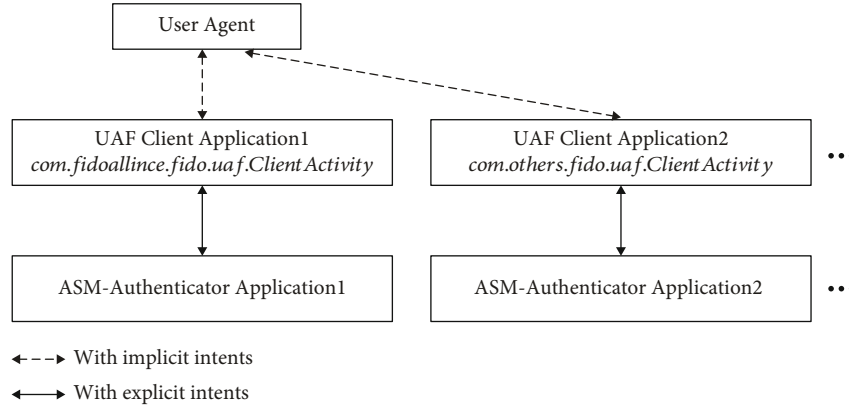


FIGURE 4: UAF implementation in Out-App Authenticator Mode.

or using TEE. We call such an application ASM-Authenticator Application.

Figure 4 describes the UAF implementation of Out-App Authenticator Mode; the specific process is as follows:

- (1) As shown in Figure 4, the User Agent starts an Activity component of the UAF Client Application with implicit intents and uses them to pass the registration or authentication request. The Android system can automatically match the intent-filter of Activity components with the intent parameters. When multiple Activity components are matched, the user will be prompted to select one of them to start. The intent-filter of an Activity component in the UAF Client is defined in Figure 5. Implicit intents enable User Agents to call multiple UAF Client Applications
- (2) After the related Activity component in the UAF Client Application is started by the User Agent, the Activity component calls *getCallingActivity()* function to obtain the caller's package name, calculates the hash of the signature certificate of the application corresponding to this package name, and generates the *FacetID* of the caller. Then, the *FacetID* is checked with *AppID*
- (3) The UAF Client Application sends the request to the ASM-Authenticator Application by starting the Activity component with explicit intents, which means that such UAF Client Application explicitly specifies the ASM-Authenticator Application to call.
- (4) After receiving the FIDO Client Application request, the ASM-Authenticator Application calculates the *CallerID* of FIDO Client Application. The calculation method is the same as that of *FacetID*. The ASM-Authenticator Application then verifies whether the caller is a valid FIDO Client Application by checking a whitelist. If the verification fails, the operation is aborted. Otherwise, the UAF Authenticator with the native implementation is called by the JNI mechanism to perform the FIDO operation

```
<intent-filter>
  <action android:name="org.fidoalliance.intent.FIDO_OPERATION" />
  <category android:name="android.intent.category.DEFAULT" />
  <data android:mimeType="application/fido.uaf_client+json" />
</intent-filter>
```

FIGURE 5: Intent-filter exposed by a UAF Client.

**3.2. In-App Authenticator Mode.** In the In-App Authenticator Mode, the UAF Client, UAF, ASM, and UAF Authenticator modules are implemented internally inside the User Agent. For example, Jingdong Finance, a financial and third-party payment application launched by Jingdong [19], implements the UAF protocol in this mode. Such applications generally implement the UAF protocol by integrating the FIDO UAF SDK that includes the above modules. Different FIDO UAF SDKs have different implementation details, but the modules and calling processes implemented in these SDKs conform to the FIDO UAF framework described by UAF protocol specification.

We summarize the implementation of a typical In-App Authenticator Mode as shown in Figure 6. UAF Client and UAF ASM send parameters by calling the interface method of the next level entity, respectively; UAF ASM stores the authentication information (such as *KeyHandle*, *KeyID*, and *UserName*) of each registration operation in the SQLite database; the authenticator starts the *FingerActivity* through explicit intents to complete user authentication and other authentication functions; *FingerActivity* calls Android's fingerprint authentication service to verify the user's identity, calls the Android KeyStore to generate the *Authentication Key* and signature, and saves the *SignCounter* to SQLite. The *FacetID* and *CallerID* of this mode are generated by calculating the hash of the User Agent's signature certificate, so these two values do not authenticate the UAF Client and UAF ASM modules in the SDK.

## 4. Authenticator Rebinding Attack

In this section, we propose an attacking method called the Authenticator Rebinding Attack which enables an attacker to rebind the victims' identity to a misused authenticator, bypass the biofactor authentication of the victim's device, and initiate unauthorized payment operations. We present

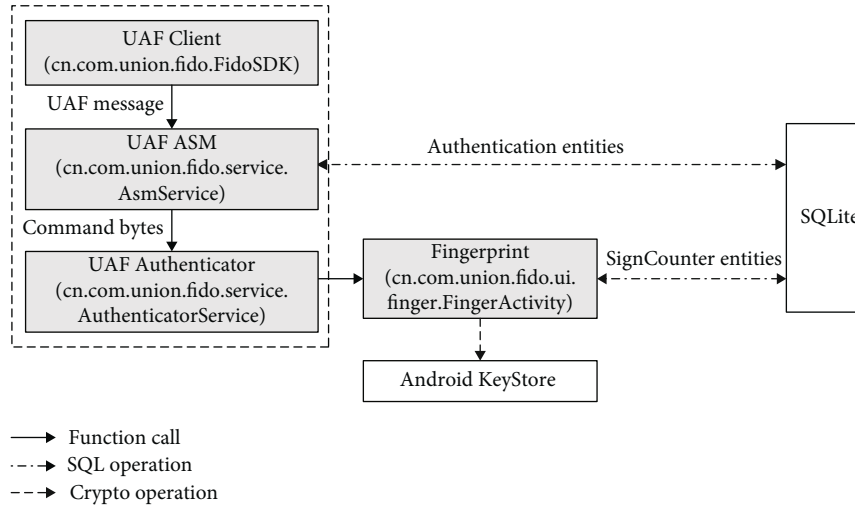


FIGURE 6: In-App Authenticator Mode.

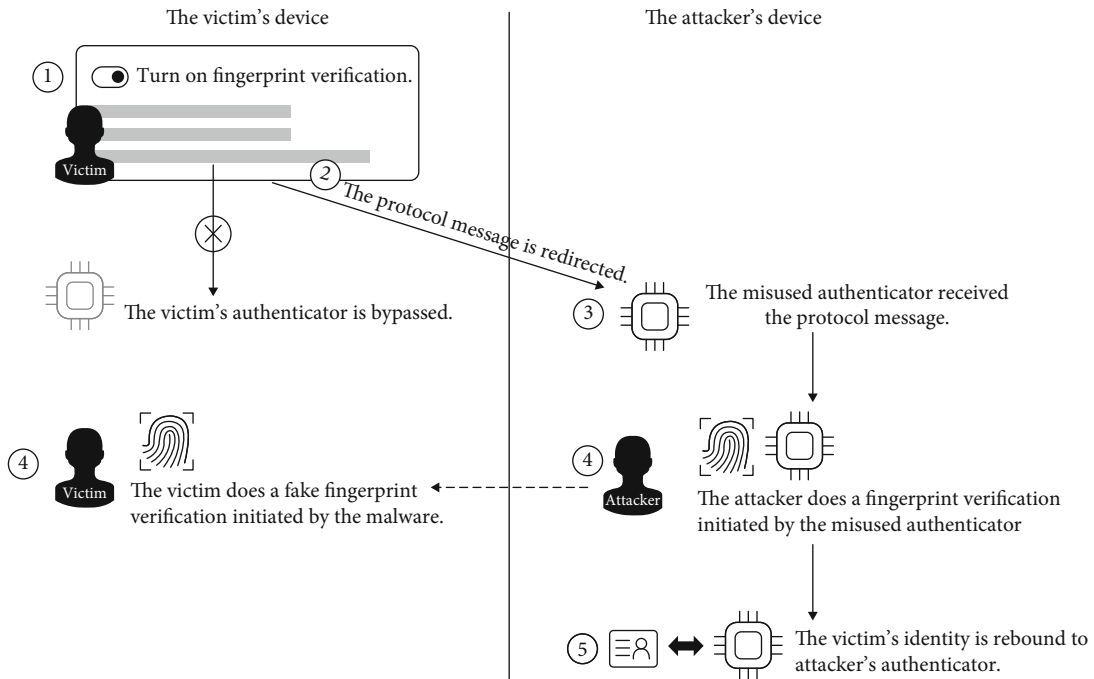


FIGURE 7: Overview of Authenticator Rebinding Attack.

the overview and details of this attack under the two implementation modes of the UAF protocol on Android, including the threat model, the attack process, and the verification of the attack on real-world applications.

4.1. *Overview of Authenticator Rebinding Attack.* Figure 7 shows an overview of the Authenticator Rebinding Attack. In the following part, we take the fingerprint authentication mechanism as a local authentication example and assume that the attacker has installed malware on the victim's device.

- (1) A victim turns on the fingerprint authentication function of an application to register a FIDO UAF service in an Android application

- (2) The malware redirects the protocol message from this application to the attacker's cracked device
- (3) The attacker tricks his/her authenticator to continue the UAF operations with the redirected message
- (4) The misused authenticator initiates a fingerprint authentication as expected. At the same time, the malware running on the victim's device uses the fake fingerprint authentication window to pretend to verify the victim's fingerprint which makes the victim not aware of any abnormalities
- (5) The attacker completes the UAF protocol registration operation on behalf of the victim and rebinds the



victim's identity to the attacker's misused authenticator. Thereafter, the attacker can bypass the fingerprint verification in the user's device and perform a transfer or payment without the user's authorization

We call this attack Authenticator Rebinding Attack because the victim's identity is eventually rebound to the attacker's authenticator. Compared with the approach using malware to steal user's passwords, this type of attack is less difficult because the attacker does not need to hack the password input window, which is always protected by the Android operating system using such techniques as TEE. This attack can be used to bypass the biometric authentication process of the FIDO UAF protocol without destroying the fingerprint verification mechanism of the Android system. Therefore, with this attack, the biometric authentication process can be bypassed in the case of remote control or temporary access to the victim's device.

We have proven that this attack is effective for both UAF protocol implementation modes, and we will present the detailed processes and verifications of such attack under different protocol implementation modes in the following sections.

**4.2. Attack under Out-App Authenticator Model.** When the User Agent of FIDO UAF is implemented using the Out-App Authenticator Mode, even if the Android operating system is not corrupted, it may suffer from an Authenticator Rebinding Attack. Meanwhile, an attacker can complete this attack at a lower cost. In this case, we call the attack Type-A Rebinding Attack.

**4.2.1. Threat Model.** In Type-A Rebinding Attack, we assume that an attacker has the following abilities.

We assume that the attacker can install malware on a victim's Android devices through system vulnerabilities, inducing users, DNS hijacking, ARP attacks, or other measures. This assumption is reasonable because the public Wi-Fi users may suffer from these attacks for the existence of Rogue Access Point (RAP) [20]. Moreover, the spread of malware is still prevalent; for example, the total number of mobile malware infections in 2018 exceeded 110 million [21]. We assume that the attacker is able to remotely control the victims' mobile device temporarily or has the opportunity to temporarily access the device without root permission. These two situations will cause the attacker to implement similar attacks using different attack schemes. For example, an attacker's malware obtains the remote control permission of the victim's device by deception, or an attacker is an acquaintance of the victim and therefore can temporarily access the phone. But in both cases, the attacker cannot replace the victim to complete the fingerprint verification process on the Android device. We also assume that the malware cannot deceive the fingerprint verification service on Android devices, because the fingerprint matching should be performed in a Trusted Execution Environment (TEE) or on a chip with a secure channel to the TEE according to the requirements of Google after Android 7.0 [22].

The attacker may crack the Android device and gain the root permission. This is necessary because the attacker has to trick the FIDO ASM-Authenticator Application in his/her own device to process the UAF protocol request forwarded from the victim's device. In fact, this can be easily satisfied for two reasons. First, many Android device vendors provide bootloader unlocking services directly or indirectly, so users can also obtain root permission by flashing a third-party ROM. Second, various automated root permission acquisition tools such as KingRoot reduce the difficulty for ordinary users to obtain root permission of the Android system. Therefore, we assume that the attacker has a device with the same model and the same software version as the victim; i.e., their FIDO ASM-Authenticator Applications have the same *AAID* and *Attestation Keys*.

**4.2.2. Processes.** Based on the above threat model, detailed attack processes of Type-A Rebinding Attack are as follows:

- (1) When a victim uses the User Agent in the user's device to open the fingerprint verification service, the registration operation of the UAF protocol is triggered to start
- (2) The User Agent obtains the FIDO UAF registration request containing *AppID* and *challenge* over the TLS channel
- (3) In Out-App Authenticator Mode, User Agent launches an Activity component of the UAF Client Application via implicit intent. The intent contains the FIDO UAF registration request
- (4) As shown in Figure 8, the Attack Agent Client and UAF Client Application expose the same intent-filter as described in Section 3.1. Therefore, the Android operating system will prompt the victim to select a UAF Client Application in the user's device for further operation by a pop-up window as shown in Figure 9
- (5) It is difficult for the victim to manually select the correct UAF Client from multiple UAF Client Applications that match implicit intents because the UAF protocol works under User Agents and is usually transparent to users. Therefore, the victim may choose the Attack Agent Client by mistake to perform further operations
- (6) Through network communication, the Attack Agent Client forwards the FIDO UAF registration request to Attack Agent Server running on the attacker's device and performs a fake fingerprint verification operation, waiting for the registration response message returned by Attack Agent Server
- (7) On the attacker's device, the Attack Agent Server passes the received FIDO UAF registration request to the ASM-Authenticator Application. Since the signature certificate of the Android application is packaged and published with the APK file, the

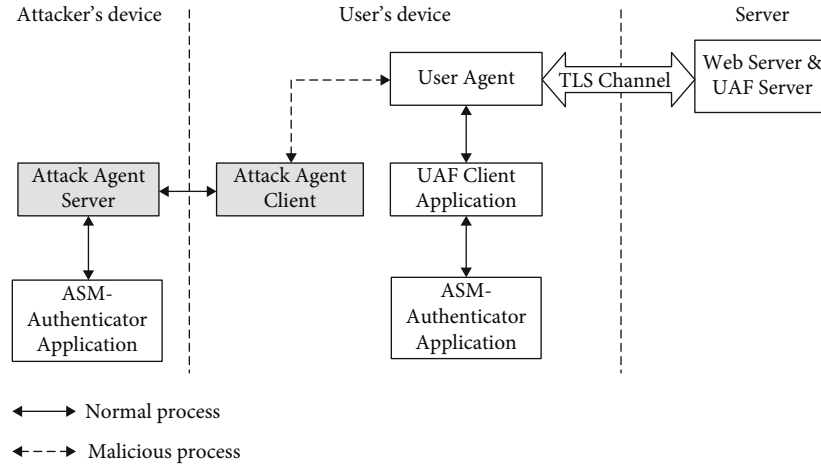


FIGURE 8: Type-A Rebinding Attack.



FIGURE 9: A pop-up window asking the victim to choose a UAF Client.

*FacetID* and *CallerID* can be easily forged. The Attack Agent Server changes the *FacetID* and *CallerID* to the correct value and then passes the modified parameters to the ASM-Authenticator Application

- (8) The ASM-Authenticator Application verifies the UAF Client Application by *CallerID*, uses the system fingerprint verification service to verify the attacker's fingerprint, and calculates the response with the *Attestation Key*. Since *CallerID* and *FacetID* are calculated in the same way and the attacker also has the root permission of the device, *CallerID* can be changed into a correct *CallerID* easily. However, it may not be necessary in cases such as the attack example described below

- (9) The registration response message generated by the misused ASM-Authenticator Application is returned to the User Agent running on the victim's device step by step according to the above path

- (10) After the victim enters his/her payment password in the User Agent for confirmation, he/she completes the registration operation of the UAF protocol using the attacker's authenticator. Thereafter, the attacker can bypass the fingerprint verification through the Attack Agent Client on this victim's device and complete the payment operations

4.2.3. *Validation*. We choose Hebao Pay as the attack target to verify the effectiveness of the Type-A Rebinding Attack. One reason for our choice is that Hebao Pay is widely used, and the cumulative number of total downloads of Hebao

Pay in China has surpassed 129 million by the end of November 2019 [23]. Another reason is that Hebao Pay uses Out-App Authenticator Mode to provide users with fingerprint verification services based on the UAF protocol. In Huawei's smart mobile devices, Hebao Pay calls system applications UAF Client and UAF ASM in EMUI (Emotion UI) to complete the UAF protocol flow. Through reverse analysis, we find that UAF ASM in EMUI includes the functions of ASM and authenticator, so it can correspond with the ASM-Authenticator Application in the above descriptions.

We implement two attack modules: Attack Agent Client and Attack Agent Server. The former exposes the same intent-filter and sets the application name and application icon similar to the UAF Client in the victim's device. The latter is achieved by using the hook methods to modify the return value of the `Activity.getCallingActivity()` function of the UAF Client in the victim's device.

In our implementation, Hebao Pay is installed on the same device with the Attack Agent Server and the return value of the `Activity.getCallingActivity()` function is changed to the package name of Hebao Pay so that UAF Client Application can always calculate the *FacetID* of Hebao Pay. The Attack Agent Client can also calculate the caller's *FacetID* and pass it to the Attack Agent Server; then, the Attack Agent Server can modify the return value of the *FacetID* calculating function to the received *FacetID*. This could make such an attack applicable to other User Agents of Out-App Authenticator Modes.

Based on the above work, we simulate the entire process of such an attack. First, the victim attempts to open the fingerprint verification service in Hebao Pay according to the described operation in the previous sections. The fingerprint verification window pops up on the screen of the attacker's mobile phone instead of the victim's phone. After the attacker performs fingerprint verification, the victim's Hebao Pay application jumps directly to the payment password input screen. The victim inputs his/her payment password to confirm this operation, and the fingerprint verification service is successfully opened. The attacker can then perform a transfer operation, and the fingerprint verification window pops up again on the screen of the attacker's mobile phone. After verifying the attacker's fingerprint, the transfer operation is successful, which means that Type-A Rebinding Attack can bypass the fingerprint verification mechanism of Out-App Authenticator Mode as expected.

**4.3. Attack under In-App Authenticator Mode.** Compared with the Type-A Rebinding Attack, the attack in the In-App Authenticator Mode that is called Type-B Rebinding Attack has the same impact on the victim but requires a higher cost. This is caused by the fact that the Relying Party function modules and authenticator in In-App Authenticator Mode are highly coupled, which prevents the User Agent from calling multiple UAF Clients, thus reducing the attack surface and increasing the difficulty of such attacks.

**4.3.1. Threat Model.** We assume that the attacker has the ability to download the User Agent and reverse the source code of the UAF protocol so that the attacker can find the attack

point at which he can redirect protocol messages in an application by manually analyzing the UAF protocol source code. It is also assumed that the malware is installed on the victim's device by the attacker and can obtain the root permission of the target device to inject the malicious code into the User Agent because the UAF protocol module of this mode is implemented inside the Reply Party Application. It also means that the attacker is able to remotely control the victims' mobile device with the root permission. The attacker is assumed to run the same In-App Authenticator Mode application on his/her cracked device, inject the malicious code, and use it as a tool to complete this attack.

**4.3.2. Processes.** According to the above threat model, the attack processes of Type-B Rebinding Attack are as follows. Steps (1) and (2) are the same as those of Type-A Rebinding Attack. (3) The attacker uses the malware to inject the malicious code into the victim's application, hook key functions related to the UAF protocol, and obtain the protocol messages. This operation requires root permissions of the victim's device. (4) The malware redirects the protocol message to the attacker's device through network communication. At the same time, the malware displays a fake fingerprint verification window to mislead the victim to wait until it receives the response from the attacker's device. (5) The broken In-App Authenticator Mode application on the attacker's device receives the protocol message and calls its authenticator mode to verify the attacker's fingerprint to generate the registration response message. (6) The broken In-App Authenticator Mode application sends back the registration response message to the victim's device. The following step is the same as step (10) in the Type-A Rebinding Attack.

**4.3.3. Validation.** We choose Jingdong Finance as the representative application of In-App Authenticator Mode to validate such attack. As of November 2019, its cumulative number of total downloads in China has exceeded 730 million [24]. Jingdong Finance implements the UAF protocol in In-App Authenticator Mode and introduces the third-party library `http://cn.com.union.fido` to implement this protocol. This library is also referenced by many other UAF applications in the In-App Authenticator Mode.

Through the reverse analysis, we find that a function named *process* is the entry function for the UAF ASM module to call the authenticator module. The parameters and return values are byte arrays. We hook this function and inject the code of parameters forwarding to implement the Attack Client and Attack Service modules. The function of the malicious code injected is shown in Figure 10, in which the *process* function is replaced by the *processHook* function and the parameters are forwarded to the remote Attack Server module. The Attack Server module is implemented by replacing this function to receive Attack Client's forwarded parameters.

Based on the above analysis, after the victim enables the fingerprint payment function in the Jingdong Finance application, the registration and authentication requests of the UAF protocol are forwarded to the attacker's device and the fingerprint verification mechanism of Jingdong Finance

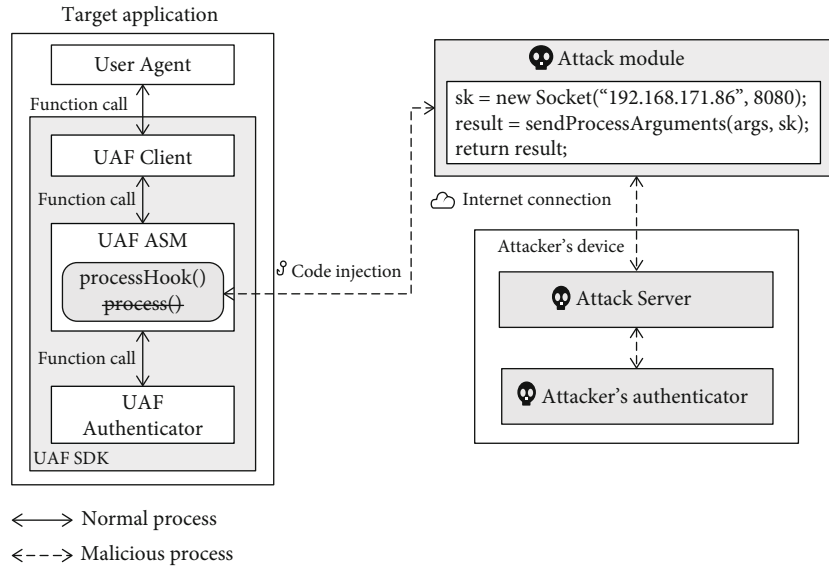


FIGURE 10: Injecting the malicious code to the target User Agent.

running on the victim's device is successfully bypassed. Despite requiring more rigorous attack conditions, Type-B Rebinding Attack is possible to happen in In-App Authenticator Mode User Agents.

**4.4. Comparison of These Two Attacks.** Both attacks under different UAF protocol implementation modes may lead to the fingerprint authentication mechanism of User Agent Applications running on the victim device to be bypassed. In general, the Type-A Rebinding Attack is easier to be implemented because the attacker does not need to obtain the root permission of the victim's device or perform a reverse analysis of the target User Agent. Moreover, some User Agents may become the potential targets during the attack because they communicate with the UAF Clients in the same way (implicit intent). However, Type-B Rebinding Attack is not easy to detect because it can be carried out without any extra interaction with the victim. Table 1 shows the difference between these two attacks.

## 5. Discussions

In this section, we first analyze the impact scope of this threat by studying the security of related applications in the actual system; then, we present its main causes and finally provide possible countermeasures that will remedy the threats.

**5.1. Impact Scope.** We manually analyze several applications that use the UAF protocol, find their characteristics, and develop programs to automatically mine such applications from a large number of Android applications. As what is claimed in the UAF protocol, if an Android application calls other UAF Client Applications to complete the FIDO UAF operation, it must declare the FIDO-related permissions in its Android manifest file [25]. Therefore, FIDO-related permissions in the manifest file can be used for searching Out-App Authenticator Mode applications. However, the applica-

tion code in the In-App Authenticator Mode does not contain the code that implements the UAF protocol but uses a third-party Java library that implements the UAF protocol instead. We automatically mine the target application by retrieving the package name and critical component name of the third-party libraries contained in an application and checking whether these names contain the FIDO keywords.

Altogether, we find 42 FIDO UAF applications in Out-App Authenticator Mode and In-App Authenticator Mode. The total download number of these 42 applications in app markets is more than 222.9 million by the end of 2019. Among these 42 applications, 8 (19%) applications call third-party UAF Client Applications (Out-App Authenticator Mode), while the remaining 34 (81%) applications use the In-App Authenticator Mode to complete the operation of the UAF protocol.

For the UAF applications in Out-App Authenticator Mode, we confirm with manual analysis methods that they all use implicit calls to interact with third-party UAF Client Applications, which means that the Type-A Rebinding Attack is effective for these applications. Even if these applications use code obfuscation and packing protections, they still cannot resist such a threat. The total downloads of these applications as shown in Table 2 have exceeded 27.1 million by far.

For the UAF applications in In-App Authenticator Mode, if users use these applications on Android devices that leak root permissions, they may become the target of Type-B Rebinding Attack. These applications are protected by code obfuscation technology for the code of the UAF protocol, and their critical method names are randomly replaced with different strings. Therefore, although attackers can determine from the package names what kind of third-party FIDO UAF libraries that the developers have used, the attackers have to manually analyze the obfuscated code of every kind of applications to find the possible hook point. This will undoubtedly increase the difficulty of carrying out this attack. Table 3

TABLE 1: The difference between the two kinds of attacks.

	Type-A Rebinding Attack	Type-B Rebinding Attack
Attack target	Some User Agents calling third-party UAF Clients	A specific User Agent with In-App Authenticator
Requiring the root permission	No	Yes
Requiring additional user interaction	Yes	No
Requiring reverse analysis	No	Yes

TABLE 2: Out-App Authenticator Mode applications.

Package name	Category	Interaction method	Downloads (million)	Attack effectiveness
com.ecitic.bank.mobile	Bank	Implicit intents	14.59	√
com.bankcomm.maidanba	Bank	Implicit intents	5.38	√
cn.com.cmbc.newmbank	Bank	Implicit intents	2.32	√
com.cmbc.cc.mbank	Bank	Implicit intents	2.32	√
com.forms	Bank	Implicit intents	0.86	√
com.cmcc.hebao	Third-party payment	Implicit intents	0.75	√
com.unicom.wopay	Third-party payment	Implicit intents	0.49	√
com.hsbank.mobilebank	Bank	Implicit intents	0.39	√

shows the third-party library package names and total downloads of the In-App Authenticator Mode applications. The attack effectiveness of third-party library `cn.com.union.fido` is confirmed in our attack validation stage, and the attack effectiveness of other libraries stays unconfirmed.

By analyzing the applications that use the UAF protocol, we can conclude that the Authenticator Rebinding Attack has already caused substantial threats to applications with a large number of downloads, especially the applications of Out-App Authenticator Mode with implicit calls.

**5.2. Main Causes.** The authentication between FIDO UAF entities is not effectively implemented in both modes. Invalid authentication between FIDO UAF entities will cause the UAF Authenticator to be abused by attackers and become an attacker’s tool for the attack. In Out-App Authenticator Mode, UAF Client Application authenticates User Agent via *FacetID* and ASM-Authenticator Application authenticates UAF Client Application via *CallerID*. As an example of our research, both *FacetID* and *CallerID* are obtained by calculating the hash of the target application’s signature certificate. However, the signature certificate can only guarantee the integrity of the Android application static code or APK file and cannot guarantee the integrity of the application at runtime. Similarly, in In-App Authenticator Mode, *FacetID* and *CallerID* cannot be used to ensure that the internal modules of a User Agent are not tampered by an attacker at runtime. Therefore, *FacetID* and *CallerID* cannot be used in these situations to guarantee the authentication between UAF protocol entities. On the contrary, if entities are effectively authenticated and the authentication information is included in the response, at least the remote server can detect whether the integrity of some entities has been compromised and then abort the protocol operation. In conclusion, it is the lack of effective authentication between entities in the imple-

mentations of the UAF protocol that the UAF protocol used in the actual system is vulnerable to the Authenticator Rebinding Attack.

**5.3. Countermeasures.** We now discuss possible countermeasures to effectively mitigate Authenticator Rebinding Attack from the perspective of protocol designers, developers of the User Agent Applications, and mobile device providers and users.

For designers of the UAF protocol, our suggestion is to enhance the authentication mechanism between the UAF entities by adding the verification of Android platform integrity based on TEE or hardware. Although the Android operating system has an isolation mechanism for applications, Android applications, for example, the application of the User Agent or the UAF Client, may still be damaged at runtime when the Android operating system is corrupted, which leads to the attack mentioned above. Therefore, if the FIDO server can authenticate the integrity of the Android operating system and combine this with the verification mechanism of *FacetID* and *CallerID*, the authentication between FIDO UAF entities can be indirectly guaranteed. For example, the TrustZone-based Integrity Measurement Architecture (TIMA) proposed by Samsung can prove the applications running in a trusted environment to the remote server [26]. And this technology can be integrated with the UAF protocol so that the authenticator can sign the *challenge* along with the attestation data, which contains boot component cryptographic hashes to indicate the integrity of the operating system. In this way, the server can determine whether the authenticator is running in a secure device by checking the TIMA attestation data.

For the developers of User Agent Applications, we first suggest using explicit intent to call the third-party UAF Client. In this case, the Package Manager Service (PMS) of the

TABLE 3: In-App Authenticator Mode libraries and applications.

Library package name	Attack effectiveness	Application package name	Code protection measure	Downloads (million)
cn.com.union.fido	√	com.jd.jrapp	Code obfuscation	23.83
		com.csii.sns.ui	App reinforcement	0.80
		com.cebbank.mobile.cemb	App reinforcement	0.36
		cn.com.bhbc.mobilebank.per	App reinforcement	0.30
		com.chinamworld.klb	App reinforcement	0.06
		cn.com.gdbank.direct	App reinforcement	0.01
		com.csii.ly.ui	App reinforcement	0.01
		com.csii.wjnsbank	App reinforcement	Less than 0.01
		com.urthinker.langfangbank.lfbank	App reinforcement	Less than 0.01
		com.csii.yk.ui	App reinforcement	Less than 0.01
com.csii.zbdirect	App reinforcement	Less than 0.01		
com.daon.fido.client.sdk	Unconfirmed	com.bochk.com	Code obfuscation	0.05
com.fido.android.framework	Unconfirmed	com.chinatelecom.bestpayclient	App reinforcement	34.45
com.iss.sdpersonalbank.fidofinger	Unconfirmed	com.iss.weifangbank	App reinforcement	0.17
		com.iss.rizhaobank	App reinforcement	0.13
		com.uccb.mobile	App reinforcement	0.13
		com.iss.changanbank	App reinforcement	0.12
		com.iss.weihaibank	App reinforcement	0.10
		com.iss.qilubank	App reinforcement	0.09
		com.iss.qishangbank	App reinforcement	0.09
		com.iss.jiningbank	App reinforcement	0.08
		com.iss.taianbank	App reinforcement	0.08
		com.iss.dongyingbank	App reinforcement	0.07
		com.iss.laishangbank	App reinforcement	0.07
		com.iss.ysantaibank	App reinforcement	0.07
		com.iss.dezhoubank	App reinforcement	0.06
		com.iss.zaozhuangbank	App reinforcement	0.02
com.lenovo.fido.framework	Unconfirmed	com.baidu.wallet	App reinforcement	1.69
		com.bill99.kuaiqian	App reinforcement	1.58
Unknown	Unconfirmed	com.icbc	App reinforcement	69.67
		com.chinamworld.bocmbci	App reinforcement	38.06
		com.icbc.im	App reinforcement	22.57
		com.baixin.mobilebank	App reinforcement	0.52
		com.icbc.collegestudents	App reinforcement	0.11

Android system can accurately locate the real UAF Client, so the malicious UAF Client hence has no chance to launch an attack. Second, the developers should consider implementing the verification mechanism to the third-party UAF Client in their applications (e.g., verifying the hash value of the third-party FIDO UAF signing certificate with a whitelist). Moreover, if the UAF protocol is implemented in In-App Authenticator Mode, application reinforcement and code obfuscating technology can be used to prevent static analysis of the applications. Finally, the hook detection mechanism [27] may also be applied so that when the attacker tries to hook functions related to the UAF protocol as described in Section 4.3, the FIDO UAF service can be disabled in time, which can prevent Type-B Rebinding Attack.

For mobile device providers, besides protecting the authenticator, a strict root detection mechanism also supported by TEE [28] should be used to protect the FIDO UAF components, which will not be compromised by malicious codes without hardware-based protections. Once it is detected that the FIDO UAF components have been corrupted, disabling the FIDO UAF service can prevent the device from being exploited by attackers in the manner shown in Section 4.2.

For users, when choosing from multiple UAF Clients, they should be careful and confirm the source and security of UAF Client; for example, check whether the UAF Client is a system application; if not, then refuse to install to make the malware difficult to disguise as a system application

without the root permission. Besides, the user should avoid using FIDO UAF authentication when the root permission of the Android device is leaked, because the malware can easily use the root permission to launch this attack silently (without additional user interaction).

## 6. Conclusions

In this paper, we analyze a novel attack named Authenticator Rebinding Attack of the UAF protocol, which makes the victim's identity be rebound to the attacker's authenticator so that the attacker can impersonate the victim's identity. In order to comprehensively study the threats of such an attack, we first analyze the applications related to third-party payment, banking, and online shopping; mine those applications that use the UAF protocol; and model two main implementations of the UAF protocol, i.e., Out-App Authenticator Mode and In-App Authenticator Mode. We then describe the detailed attack process of these two implementation modes. We also demonstrate that the proposed attacks do work by performing attack verification on typical actual applications. Besides, the applications that use UAF protocol on the Android platform in the actual system are threatened by this attack and the applications that make implicit calls in Out-App Authenticator Mode are more vulnerable. This threat can be attributed to the lack of effective authentication between entities when the UAF protocol is implemented on the Android platform. The *FacetID* and *CallerID* used by the UAF protocol cannot prove the integrity of the User Agent and UAF Client. We finally present countermeasures that can prevent this threat. We believe that our research on the Authenticator Rebinding Attack of the UAF protocol can help protocol designers, User Agent Application developers, and mobile device providers and users to improve the security of the UAF protocol.

## Data Availability

The APK files used to support the findings of this study are downloaded from <http://zhushou.360.cn/>. The python script used to support the findings of this study is uploaded to the git repository <https://github.com/PandaQ2014/FindFIDO>. The statistical data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

This research is supported by the National Science and Technology Major Project of China (2018ZX03001010-005).

## References

- [1] S. Machani, R. Philpott, S. Srinivas, J. Kemp, and J. Hodges, *FIDO UAF Architectural Overview*, FIDO Alliance, 2017.

- [2] FIDO Alliance, "FIDO certified products," 2019, <https://fidoalliance.org/certification/fido-certified-products/>.
- [3] K. Hu and Z. Zhang, "Security analysis of an attractive online authentication standard: FIDO UAF protocol," *China Communications*, vol. 13, no. 12, pp. 189–198, 2016.
- [4] C. Xenakis, C. Panos, S. Malliaros, C. Ntantogian, and A. Panou, *A security evaluation of FIDO's UAF protocol in mobile and embedded devices*, International Tyrrhenian Workshop Springer, Cham, 2017.
- [5] R. Lindemann, D. Baghdasaryan, and B. Hill, *FIDO security reference*, FIDO Alliance Proposed Standard, 2015.
- [6] FIDO Alliance, "Certification Overview," 2019, <https://fidoalliance.org/certification/>.
- [7] International Data Corporation, "Smartphone market share," 2020, <https://www.idc.com/promo/smartphone-market-share/vendor>.
- [8] Y. Zhang, X. Wang, Z. Zhao, and H. Li, "Secure display for FIDO transaction confirmation," in *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, pp. 155–157, New York, NY, USA, 2018.
- [9] StatCounter, "Mobile operating system market share worldwide," 2020, <https://gs.statcounter.com/os-market-share/mobile/worldwide>.
- [10] FIDO Alliance, "FIDO certified showcase," 2019, ). <https://fidoalliance.org/fido-certified-showcase>.
- [11] "FIDO Alliance FIDO UAF architectural overview," 2017, <https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-uaf-overview-v1.1-id-20170202.html>.
- [12] M. Dietz, A. Czeskis, D. Balfanz, and D. S. Wallach, "Origin-bound certificates: a fresh approach to strong client authentication for the web," in *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*, pp. 317–331, Bellevue, WA, 2012.
- [13] FIDO Alliance, "FIDO UAF protocol specification," 2017, <https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-uaf-protocol-v1.1-id-20170202.html>.
- [14] R. Lindemann, E. Tiffany, B. Davit, D. Balfanz, B. Hill, and J. Hodges, *FIDO UAF protocol specification v1.1*, FIDO Alliance, 2017.
- [15] FIDO Alliance, "FIDO UAF authenticator-specific Module API," 2017, <https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-uaf-asm-api-v1.1-id-20170202.html>.
- [16] FIDO Alliance, "FIDO AppID and Facet specification," 2017, <https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-appid-and-facets-v1.1-id-20170202.html>.
- [17] FIDO Alliance, "FIDO technical glossary," 2017, <https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-glossary-v1.1-id-20170202.html>.
- [18] China Mobile, *Hebao Pay, pay for reliability*, China Mobile Limited, 2020, <https://www.cmpay.com/>.
- [19] JD Digits, *A Friend Who Understands Finance*, JD Digits, 2020, <https://jr.jd.com/>.
- [20] W. Yang, X. Li, Z. Feng, and J. Hao, "TLSsem: a TLS security-enhanced mechanism against MITM attacks in public WiFi," in *2017 22nd International Conference on Engineering of Complex Computer Systems (ICECCS)*, Fukuoka, Japan, 2017.
- [21] Beijing Qihu Keji Co Ltd, *2018 Android Malware Special Report*, Technical Report, 2018.
- [22] Google Inc, "Android compatibility definition (Android 7.0)," 2017, <https://source.android.google.cn/compatibility/7.0/android-7.0-cdd>.

- [23] Kuchuan, “Hebao payment application data page,” 2019, <https://android.kuchuan.com/page/detail/download?package=com.cmcc.hebao&infomarketid=10&site=0#!/sum/com.cmcc.hebao>.
- [24] Kuchuan, “Jingdong Finance application data page,” 2019, <https://android.kuchuan.com/page/detail/download?package=com.jd.jrapp&infomarketid=1&site=0#!/sum/com.jd.jrapp>.
- [25] B. Hill, D. Baghdasaryan, B. Blanke, J. Hodges, and K. Yang, *FIDO UAF application API and transport binding specification v1.1*, FIDO Alliance, 2017.
- [26] A. M. Azab, P. Ning, J. Shah et al., “Hypervision across worlds: real-time kernel protection from the ARM TrustZone secure world,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*, pp. 90–102, New York, NY, USA, 2014.
- [27] M. Szczepanik, I. J. Józwiak, P. P. Józwiak, M. Kędziora, and J. Mizera-Pietraszko, “Android hook detection based on machine learning and dynamic analysisWeb, Artificial Intelligence and Network Applications,” vol. 1150 of WAINA 2020. *Advances in Intelligent Systems and Computing*, Springer, Cham, 2020.
- [28] GlobalPlatform, *The trusted execution environment: delivering enhanced security at a lower cost to the mobile market*, GlobalPlatform Inc, 2015.